

# Analiza Obrazów

## Sprawozdanie I

Yuliya Zviarko

Wydział Fizyki i Informatyki Stosowanej  
Akademia Górnictwo-Hutnicza im. Stanisława Staszica w Krakowie

11 listopada 2024

# Spis Treści

## Spis treści

<b>1 Wprowadzenie</b>	<b>3</b>
1.1 Podstawy Grafiki Rastrowej . . . . .	3
1.2 Podstawowe Operacje na Obrazach w Języku Matlab . . . . .	4
<b>2 Laboratorium 2</b>	<b>6</b>
2.1 Wyświetlanie kanałów RGB . . . . .	7
2.2 Histogramy dla kanałów RGB . . . . .	8
2.3 Konwersja do skali szarości . . . . .	8
2.4 Zmiana jasności, kontrastu i korekcja gamma . . . . .	9
2.4.1 Zwiększenie jasności . . . . .	9
2.4.2 Zmiana kontrastu . . . . .	11
2.4.3 Korekcja gamma . . . . .	12
2.4.4 Wyrównanie histogramu . . . . .	13
<b>3 Laboratorium 3 - Filtry</b>	<b>15</b>
3.1 Opis sąsiedztwa pikseli . . . . .	15
3.2 Filtr uśredniający (Blur) . . . . .	15
3.3 Filtr z różnymi wagami (Gaussian Blur) . . . . .	16
3.4 Rozmycie w kierunku pionowym . . . . .	17
3.5 Użycie wag ujemnych . . . . .	18
3.6 Filtr krawędziowy . . . . .	19
3.7 Filtr medianowy . . . . .	20
3.8 Binaryzacja . . . . .	21
<b>4 Laboratorium 4 - Transformacja Fouriera</b>	<b>23</b>
4.1 Obliczenie transformacji Fouriera . . . . .	24
4.2 Rekonstrukcja obrazu . . . . .	25
4.3 Kompresja obrazu . . . . .	27

# 1 Wprowadzenie

## 1.1 Podstawy Grafiki Rastrowej

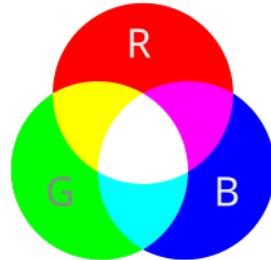
**Grafika rastrowa**, znana również jako **bitmapa**, jest sposobem przedstawiania obrazów w postaci prostokątnej siatki pikseli, które stanowią najmniejsze jednostki obrazu. Technika ta jest szczególnie użyteczna przy zapisie zdjęć oraz realistycznych obrazów, gdzie istotne są subtelne przejścia tonalne i szczegółowe. Grafika rastrowa charakteryzuje się kilkoma istotnymi cechami:

- **Wymiary obrazu:** Obrazy rastrowe opisuje się za pomocą wymiarów, czyli liczb pikseli w szerokości i wysokości.  
Przykładowo, obraz o wymiarach 1920x1080 pikseli oznacza 1920 pikseli szerokości i 1080 pikseli wysokości.
- **Model kolorystyczny:** Każdy piksel może być reprezentowany w różnych modelach kolorystycznych. Najczęściej używanym modelem jest **RGB**, który składa się z trzech składowych:

Czerwony (R)

Zielony (G)

Niebieski (B)



Rysunek 1: Model kolorów RGB.

Każdy z tych kolorów reprezentowany jest przez wartość w zakresie od 0 do 255, co umożliwia uzyskanie szerokiej gamy barw.

- **Obrazy monochromatyczne:** Często dokonuje się konwersji obrazu kolorowego na monochromatyczny, co pozwala skupić się na intensywności światła, eliminując wpływ koloru. Typowe wagi dla przekształcenia RGB do skali szarości to:

- 0.299 dla koloru czerwonego,
- 0.578 dla koloru zielonego,
- 0.144 dla koloru niebieskiego.

- **Kompresja obrazu:** Obrazy rastrowe można poddawać kompresji w celu zmniejszenia ich rozmiaru pliku, co ułatwia przechowywanie i przesyłanie. Kompresja może być **stratna** (np. formaty JPEG, WebP), prowadząc do utraty części danych, lub **bezstratna** (np. formaty PNG, TIFF), zachowując wszystkie informacje obrazu. **Obrazy kompresowane** różnią się w zależności od zastosowania – w fotografii, skanach dokumentów czy ilustracjach komputerowych, gdzie realistyczne odwzorowanie kolorów nie zawsze jest konieczne.
- **Szerokie zastosowanie:** Grafika rastrowa znajduje zastosowanie w licznych dziedzinach, takich jak fotografia cyfrowa, projektowanie graficzne (banery, plakaty), a także obrazowanie medyczne (np. zdjęcia rentgenowskie, tomografia komputerowa) i wiele innych.

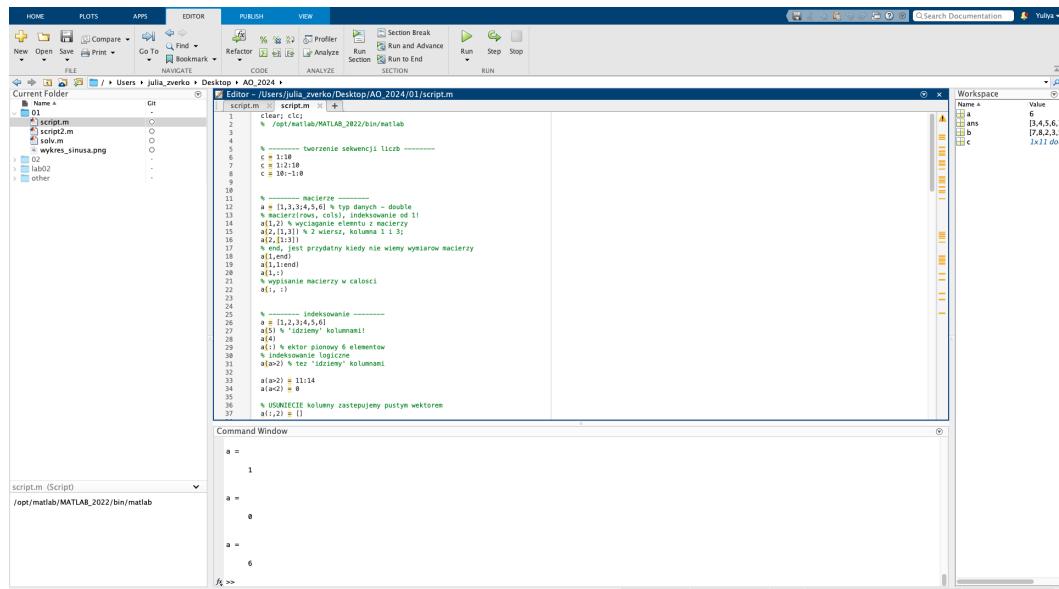
## 1.2 Podstawowe Operacje na Obrazach w Języku Matlab

Do analizy obrazów na zajęciach wykorzystaliśmy środowisko programistyczne **Matlab**, które jest popularnym narzędziem w zakresie przetwarzania i analizy obrazów. Matlab oferuje wygodne środowisko pracy oraz szeroki zbiór pakietów zawierających narzędzia i algorytmy, które znacząco przyspieszają i ułatwiają proces analizy obrazów bez konieczności stosowania zaawansowanych metod matematycznych.

Do najczęściej stosowanych funkcji należały:

- **imread** – funkcja służąca do wczytywania obrazów z plików. Jest niezbędna na etapie wstępnej analizy obrazu, umożliwiając ładowanie danych w różnych formatach.
- **imshow(<image>)** – funkcja pozwalająca na wyświetlenie obrazu `<image>`.
- **subplot** – funkcja, która umożliwia podzielenie obszaru wykresu na siatkę, co pozwala na równoczesne wyświetlanie wielu obrazów lub wykresów w jednym oknie.
- **imhist** – generuje histogram intensywności pikseli obrazu, co pozwala na analizę rozkładu jasności oraz kontrastu.
- **mean** – oblicza średnią wartość elementów macierzy.
- **permute** – funkcja umożliwiająca zmianę wymiarów macierzy w określonej kolejności.
- **rgb2gray** – funkcja konwertująca obraz kolorowy do odcieni szarości. Jest stosowana, gdy kolor nie jest istotny dla analizy, pozwalając na uproszczenie obliczeń.
- **imfilter** – stosuje zadany filtr do obrazu, co umożliwia wykonanie operacji takich jak wygładzanie, wyostrzanie, czy wykrywanie krawędzi.

- **medfilt2** – funkcja służąca do przetwarzania obrazu w skali szarości lub kolorowego obrazu poprzez zastąpienie wartości każdego piksela medianą wartości pikseli w jego otoczeniu.
- **fft2** – funkcja obliczająca dwuwymiarową transformację Fouriera obrazu. Stosowana jest przy analizie częstotliwościowej obrazu.
- **ifft2** – funkcja obliczająca dwuwymiarową odwrotną transformację Fouriera. Stosuje się ją, aby przekształcić obraz z przestrzeni częstotliwościowej z powrotem do przestrzeni czasowo-przestrzennej, czyli oryginalnej postaci obrazu.
- **abs** – funkcja obliczająca wartość bezwzględną elementów macierzy, co jest użyteczne przy analizie wyników złożonych operacji matematycznych, takich jak transformacja Fouriera.



Rysunek 2: Interfejs Matlab – narzędzie do analizy i przetwarzania obrazów.

## 2 Laboratorium 2

Podczas zajęć analizowano obraz *zubr.jpg*, który przedstawia scenę z żubrem stojącym na tle trawy. Żubr, jako główny obiekt obrazu, posiada brązowe umaszczenie z odcieniami pomarańczowego i czerwonego, natomiast trawa wokół niego jest zielona.



Rysunek 3: Oryginalny obraz żubra wykorzystywany podczas zajęć.

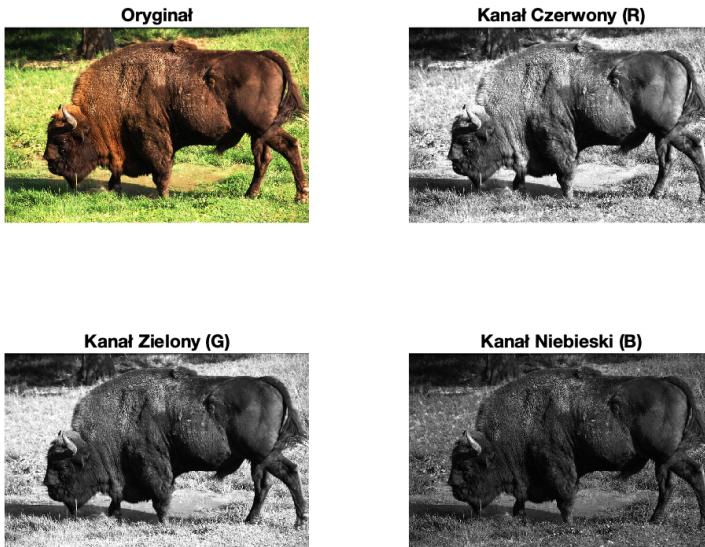
Obraz został wczytany jako macierz typu **uint8** o wartościach w przedziale **0-255** oraz wymiarach: **642x1000x3** (wysokość x szerokość x 3 kanały, odpowiadające za intensywność kolorów w przestrzeni RGB). Aby umożliwić bardziej precyzyjną analizę, obraz został znormalizowany do przedziału [0, 1], zgodnie ze standardem często stosowanym w środowisku Matlab dla obrazów typu **double**. Transformacji dokonano za pomocą poniższego kodu:

```
% Wczytanie obrazu i normalizacja do przedziału [0,1]
im = imread('zubr.jpg');
im = double(im) / 255;
```

## 2.1 Wyświetlanie kanałów RGB

Pierwszym etapem analizy było wyświetlenie poszczególnych kanałów RGB obrazu w celu dokładniejszego zbadania rozkładu kolorów:

- **Oryginał:** Centralnym elementem obrazu jest duży żubr o brązowym umaszczeniu z odcieniami pomarańczowego i czerwonego. Tło stanowi zielona trawa, co nadaje obrazowi naturalny kontrast.
- **Kanał R (czerwony):** W tym kanale żubr jest dobrze widoczny, zwłaszcza w jasnych partiach jego ciała, co sugeruje, że kolor czerwony jest intensywniej obecny w umaszczeniu żubra. Trawa jest również widoczna, lecz o niższej intensywności, co podkreśla dominującą obecność czerwieni na obiekcie głównym.
- **Kanał G (zielony):** Trawa jest również jasna, lecz widać w niej więcej szczegółów, a żubr jest ciemniejszy. Taki wynik sugeruje, że zielone barwy są dominujące w tle (trawie), podczas gdy ciało żubra zawiera mniej zielonych, co sprawia, że jest ciemniejsze.
- **Kanał B (niebieski):** Obraz jest bardzo ciemny w tym kanale, ponieważ ani trawa, ani żubr nie mają intensywnych odcieni niebieskich

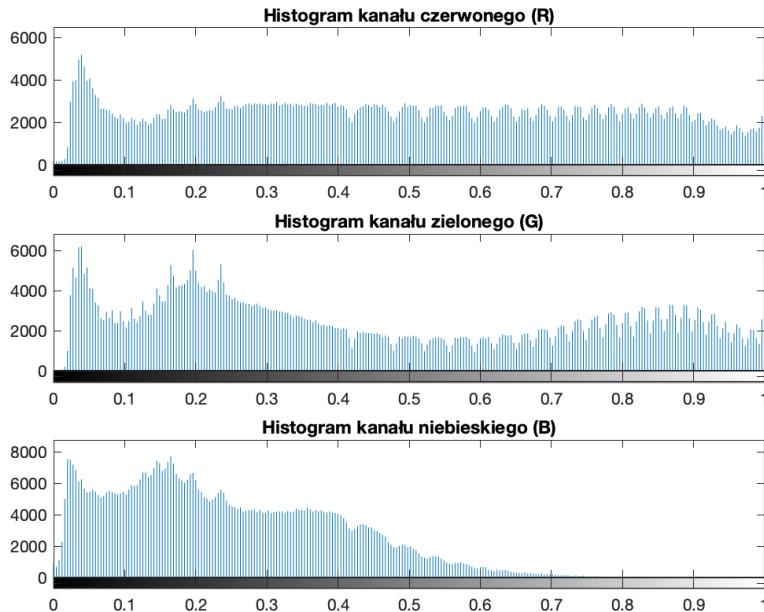


Rysunek 4: Podział obrazu na kanały RGB.

## 2.2 Histogramy dla kanałów RGB

Histogramy kanałów RGB ujawniają różnice w rozkładzie intensywności pikseli, co odzwierciedla dominację poszczególnych barw na obrazie:

- **Kanał R (czerwony):** Jednorodny rozkład intensywności, szczególnie w jaśniejszych wartościach, sugeruje obecność czerwieni głównie na żubrze i niektórych obszarach trawy.
- **Kanał G (zielony):** Dwumodalny histogram wskazuje na dominację zieleni w tle (trawa) oraz jej niewielką obecność na ciele żubra.
- **Kanał B (niebieski):** Przewaga ciemnych wartości wyjaśnia niski poziom niebieskiego koloru w obrazie, co sprawia, że jest on słabo widoczny.



Rysunek 5: Histogramy intensywności dla kanałów RGB.

## 2.3 Konwersja do skali szarości

Istnieje kilka metod konwersji:

1. **Średnia arytmetyczna (R, G, B):** Uśrednia wartości pikseli z trzech kanałów:

$$G = \frac{R + G + B}{3}$$

Jest szybka, ale nie zawsze dobrze odwzorowuje jasność percepcyjną.

2. **Największa wartość (max)**: Wybiera najwyższą wartość spośród R, G, B:

$$G = \max(R, G, B)$$

Wydobywa najjaśniejsze detale obrazu.

3. **Najmniejsza wartość (min)**: Wybiera najniższą wartość z kanałów:

$$G = \min(R, G, B)$$

Przyciemnia obraz, eksponując ciemniejsze obszary.

4. **Medianą**: Oblicza medianę z wartości R, G, B, co zwiększa odporność na skrajne wartości:

$$G = \text{median}(R, G, B)$$

5. **Metoda YUV**: Zastosowanie wag [0.299, 0.587, 0.114] pozwala na lepsze odwzorowanie jasności, co sprawia, że wynikowy obraz jest bardziej naturalny:

$$G = 0.299 \times R + 0.587 \times G + 0.114 \times B$$



Rysunek 6: Obraz w skali szarości wygenerowany metodą YUV.

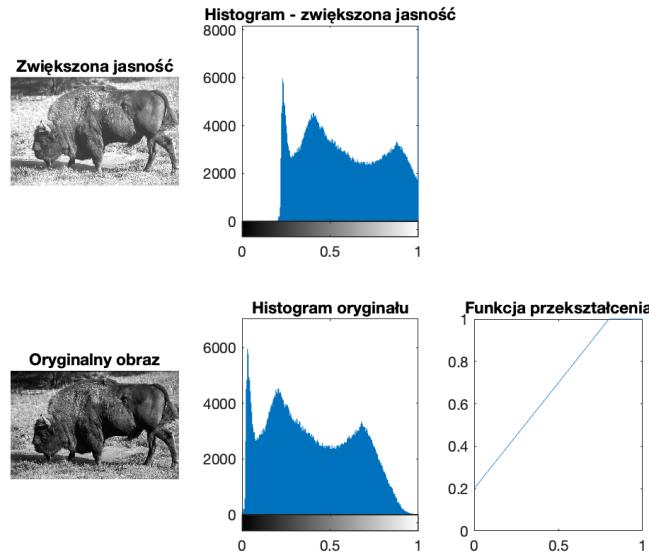
## 2.4 Zmiana jasności, kontrastu i korekcja gamma

### 2.4.1 Zwiększenie jasności

Aby poprawić jasność obrazu, zastosowano metodę, polegającą na dodaniu stałej wartości  $b = 0.2$  do intensywności wszystkich pikseli. Dzięki temu każdy piksel staje się jaśniejszy:

$$G_{\text{new}} = G_{\text{old}} + b$$

gdzie  $G_{\text{new}}$  to nowa wartość pikseli,  $G_{\text{old}}$  to oryginalna wartość pikseli, a  $b$  to dodana wartość. W przypadku, gdy nowa wartość pikseli przekracza zakres  $[0, 1]$ , jest ona ograniczana do wartości maksymalnej 1, a wartości poniżej zera są ustalane na 0, żeby obraz nie zawierał wartości poza dopuszczalnym zakresem:



Rysunek 7: Obraz po zwiększeniu jasności.

Wynikiem zastosowania tej metody jest wyraźne rozjaśnienie obrazu. Histogram nowego obrazu pokazuje, że więcej pikseli znajduje się w jaśniejszych rejonach skali szarości. Przedstawiono również funkcję przekształcenia, która pokazuje, jak wartości pikseli zostały przekształcone w wyniku dodania stałej:

$$y = x + b,$$

gdzie  $x$  to oryginalna wartość pikseli. Na wykresie funkcji przekształcenia (rysunek 7) widać, że wartości zostały przesunięte w górę o wartość  $b$ , co skutkuje zwiększeniem jasności obrazu.

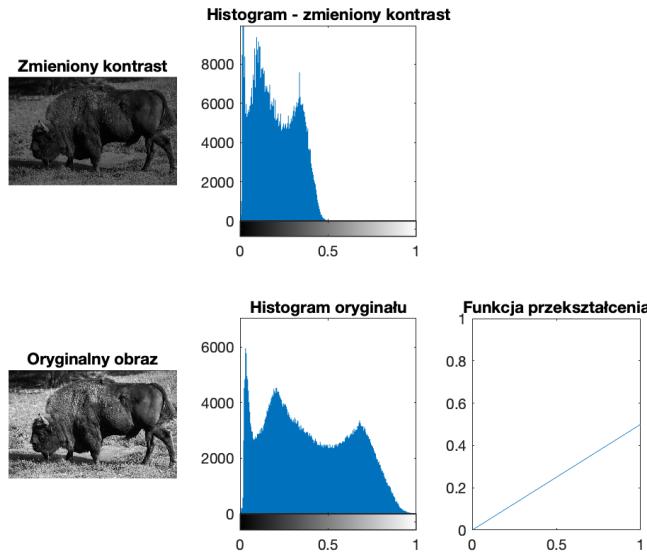
### 2.4.2 Zmiana kontrastu

Aby zmienić kontrast obrazu pomnożono intensywności wszystkich pikseli przez stałą wartość  $c = 0.5$ , co skutkowało zmniejszeniem kontrastu, ponieważ wszystkie wartości pikseli są skalowane w dół. To powoduje, że różnice między jasnymi i ciemnymi obszarami obrazu stają się mniej wyraźne.

$$C_{\text{new}} = C_{\text{old}} \times c,$$

gdzie  $C_{\text{new}}$  to nowa wartość pikseli,  $C_{\text{old}}$  to oryginalna wartość pikseli, a  $c$  to wartość skali. Podobnie jak w przypadku zwiększenia jasności, wprowadzono ograniczenia, aby upewnić się, że wartości pikseli pozostają w zakresie  $[0, 1]$ :

$$\begin{aligned} C_{\text{new}} > 1 &\Rightarrow C_{\text{new}} = 1 \\ C_{\text{new}} < 0 &\Rightarrow C_{\text{new}} = 0 \end{aligned}$$



Rysunek 8: Obraz po zmianie kontrastu.

W wyniku zastosowania tej metody obraz stał się mniej kontrastowy. Histogram obrazu ilustruje zmiany w rozkładzie intensywności pikseli, pokazując, że wartości intensywności stały się bardziej zbliżone do siebie do czarnego spektrum.

Dodatkowo, przedstawiono funkcję przekształcenia:

$$y = x \times c,$$

gdzie  $x$  to oryginalna wartość pikseli. Na wykresie funkcji przekształcenia (rysunek 8) widać, że wartości zostały przesunięte w dół, co skutkuje zmniejszeniem kontrastu obrazu.

#### 2.4.3 Korekcja gamma

W celu poprawy wyrazistości obrazu w ciemniejszych obszarach zastosowano korekcję gamma. Wartości intensywności pikseli podniesiono do potęgi  $\frac{1}{\gamma}$ , co skutkuje nieliniowym przekształceniem wartości pikseli:

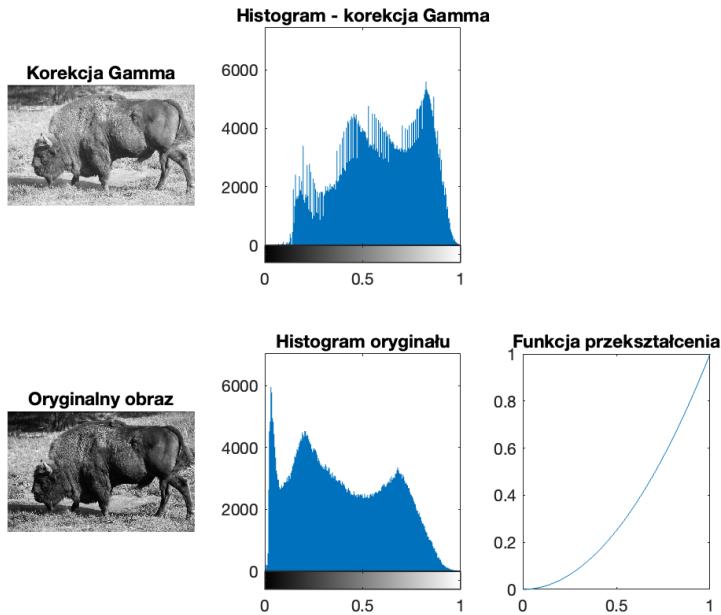
$$I_{\text{new}} = I_{\text{old}}^{\frac{1}{\gamma}},$$

gdzie  $I_{\text{new}}$  to nowa wartość intensywności,  $I_{\text{old}}$  to oryginalna wartość intensywności, a  $\gamma$  to parametr korekcji. W przypadku naszej analizy przyjęto wartość  $\gamma = 2$ :

$$I_{\text{new}} = I_{\text{old}}^{0.5},$$

To przekształcenie poprawia szczegóły w ciemnych częściach obrazu, co jest szczególnie przydatne, gdy obraz jest zbyt ciemny lub brakuje mu kontrastu w niższych tonach. Użyto również ograniczeń, aby zapewnić, że wartości pikseli pozostają w zakresie  $[0, 1]$ :

$$\begin{aligned} I_{\text{new}} > 1 &\Rightarrow I_{\text{new}} = 1 \\ I_{\text{new}} < 0 &\Rightarrow I_{\text{new}} = 0 \end{aligned}$$



Rysunek 9: Obraz po korekci gamma.

Wynikiem zastosowania korekcji gamma było zwiększenie wyrazistości w ciemniejszych obszarach obrazu, co pozwoliło na lepsze uchwycenie szczegółów w miejscach, które wcześniej były zbyt ciemne. Histogram po korekcji gamma pokazuje, że rozkład intensywności pikseli przesunął się w stronę jaśniejszych wartości, co potwierdza poprawę widoczności detali.

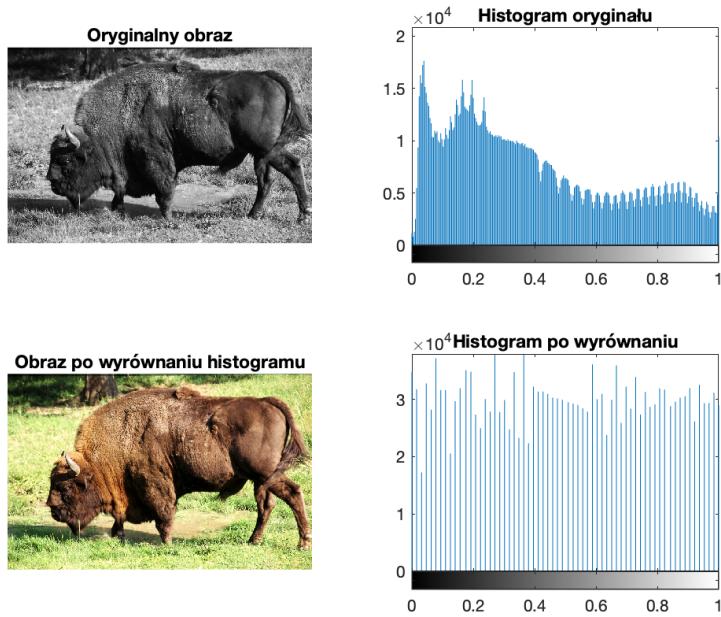
Dodatkowo, przedstawiono funkcję przekształcenia nieliniowego, gdzie  $x$  to oryginalna wartość pikseli:

$$y = x^\gamma.$$

Wykres tej funkcji pokazuje nieliniowy charakter korekcji gamma.

#### 2.4.4 Wyrównanie histogramu

Ostatecznym krokiem w przetwarzaniu obrazu było wyrównanie histogramu. Proces ten ma na celu poprawę rozkładu intensywności pikseli, co prowadzi do bardziej jednolitego obrazu. Wyrównanie histogramu działa poprzez rozłożenie intensywności pikseli w sposób, który zwiększa kontrast i wydobywa szczegóły.



Rysunek 10: Obraz po wyrównaniu histogramu.



Rysunek 11: Oryginalny obraz żubra wykorzystywany podczas zajęć przed wyrównaniem histogramu.

Na powyższym rysunku (rysunek 10) widać obraz po wyrównaniu histogramu, który pokazuje poprawiony rozkład intensywności. Histogram po wyrównaniu ilustruje, jak wartości intensywności zostały lepiej rozłożone w zakresie, co wpływa na ogólną jakość wizualną obrazu.

Przeprowadzone analizy dostarczyły cennych informacji na temat dominujących kolorów i ich rozkładu w obrazie, co może być przydatne w dalszych badaniach, które wymagają precyzyjnej analizy wizualnej.

### 3 Laboratorium 3 - Filtry

Celem tych zajęć była analiza przetwarzania obrazu na przykładzie zdjęcia żubra z poprzednich zajęć, przy użyciu różnych technik filtracji.

Obraz został wczytany, przekształcony do skali szarości i znormalizowany.

#### 3.1 Opis sąsiedztwa pikseli

Piksele sąsiadujące są definiowane jako te, które mają co najmniej jeden wspólny wierzchołek. Istnieją dwa podstawowe typy sąsiedztwa:

**Krawędź (von Neumann)** Obejmuje cztery piksele sąsiadujące w pionie i poziomie.

**Wierzchołek (Moore)** Obejmuje osiem pikseli sąsiadujących, w tym te znajdujące się na przekątnych.

W kontekście filtracji obrazów, istotne jest rozważenie różnych rzędów sąsiedztwa pikseli.

#### 3.2 Filtr uśredniający (Blur)

Zastosowanie filtra uśredniającego w celu usunięcia niepożądanych ekstremów oraz odszumieniu obrazu. Rozmiar filtru( $k$ ) wynosił 33 piksele. Filtr ten działa poprzez zamianę wartości danego piksela na średnią wartość pikseli z jego otoczenia, co prowadzi do wygładzenia obrazu.

W zastosowanej implementacji filtr uśredniający można opisać matematycznie jako:

$$f = \frac{1}{k^2} \begin{bmatrix} 1 & 1 & \dots & 1 \\ 1 & 1 & \dots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \dots & 1 \end{bmatrix} \quad (1)$$

Aplikacja filtra do obrazu przedstawia się w postaci:

$$\text{fim} = \text{imfilter}(\text{im}, f) \quad (2)$$

Zastosowanie filtra do obrazu prowadzi do redukcji szumów i wygładzenia detali:



Rysunek 12: Obraz po filtracji - Blur

Obraz po zastosowaniu filtra uśredniającego wykazuje znaczną redukcję szumów oraz wygładzenie detali, co jest przydatne w przypadku obrazów z dużą ilością zakłóceń. Należy jednak zauważyć, że intensywne wygładzanie może prowadzić do utraty ważnych informacji o krawędziach obiektów jak i w naszym przypadku.

### 3.3 Filtr z różnymi wagami (Gaussian Blur)

Filtr Gaussa stosuje różne wagi dla pikseli sąsiadujących, co pozwala uzyskać naturalniejsze rozmycie. W przeciwieństwie do filtra uśredniającego, filtr Gaussa nadaje większe znaczenie pikselom bliżej środka, co prowadzi do delikatniejszego efektu rozmycia.

Matematycznie, filtr Gaussa opisuje się jako:

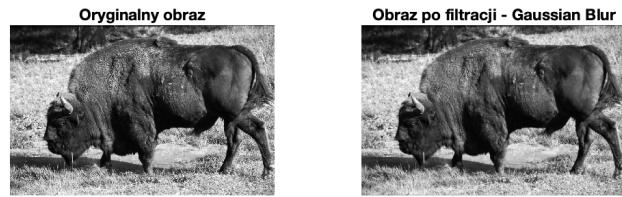
$$G(x, y) = \frac{1}{2\pi\sigma^2} e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (3)$$

gdzie  $\sigma$  to odchylenie standardowe.

Aplikacja filtru do obrazu przedstawia się w postaci:

$$\text{fim} = \text{imfilter}(\text{im}, G) \quad (4)$$

Filtr Gaussa skutecznie redukuje szумy oraz wygładza detale. Dodatkowo, zapewnia łagodniejsze przejścia tonalne, co skutkuje bardziej naturalnym efektem rozmycia, szczególnie w obszarach z miękkimi krawędziami



Rysunek 13: Obraz po filtracji - Gaussian Blur

### 3.4 Rozmocie w kierunku pionowym

W celu rozmycia obrazu w kierunku pionowym zastosowano filtr gausowski, który uwzględnia wartości w otoczeniu piksela. Filtr został zdefiniowany jako macierz  $f$ , gdzie centralna wartość jest największa (4), a sąsiednie wartości są mniejsze (2):

$$f = [000; 242; 000]/8; \quad (5)$$

Dzięki temu filtr działa w sposób, który sprawia, że wartości pikseli są wygładzane w pionie.

```
fim = imfilter(im, f);
```

Powyższa funkcja `imfilter` nakłada filtr na oryginalny obraz, co skutkuje rozmyciem:



Rysunek 14: Rozmycie w kierunku pionowym

Filtracja w kierunku pionowym prowadzi do uwydatnienia poziomych struktur w obrazie, co umożliwia lepszą identyfikację krawędzi oraz konturów obiektów.

### 3.5 Użycie wag ujemnych

W tej metodzie zastosowano filtr z wagami ujemnymi, co pozwala na uwypuklenie różnic w obrazie. Filtr został zdefiniowany jako macierz  $f$  o wartościach  $-1$ , z wyjątkiem wartości centralnej, która wynosi  $k^2$ . Dzięki temu filtr działa jak operator podkreślający różnice, co może być szczególnie przydatne w analizie konturów i drobnych szczegółów obrazu.

Macierz filtru  $f$  wygląda następująco dla  $k = 3$ :

$$f = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 9 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

```
fim = imfilter(im, f);
```

Powyzsza operacja umożliwia wyodrębnienie bardziej wyraźnych krawędzi w obrazie.



Rysunek 15: Użycie wag ujemnych do uwypuklenia różnic

Wykorzystanie wag ujemnych w procesie filtracji skutkuje wzmacnianiem kontrastów oraz uwidacznianiem różnic pomiędzy pikselami, co znacząco poprawia identyfikację detali w obrazie.

### 3.6 Filtr krawędziowy

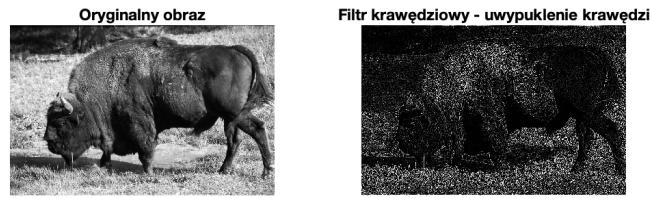
Filtr krawędziowy również używa wag ujemnych, jednak jego konstrukcja zapewnia, że suma wag wynosi zero. Wartość centralna filtra jest ustaliona na  $k^2 - 1$ , co w połączeniu z ujemnymi wagami wokół niej umożliwia wyodrębnienie krawędzi.

Macierz filtru  $f$  wygląda następująco dla  $k = 3$ :

$$f = \begin{bmatrix} -1 & -1 & -1 \\ -1 & 8 & -1 \\ -1 & -1 & -1 \end{bmatrix}$$

```
fim = imfilter(im, f);
```

Operacja ta przetwarza obraz, podkreślając obszary, w których następuje nagła zmiana intensywności.



Rysunek 16: Filtr krawędziowy - uwypuklenie krawędzi

Filtr krawędziowy efektywnie podkreśla kontury obiektów, co czyni go użytecznym narzędziem w zadaniach związanych z detekcją. Należy jednak pamiętać na wprowadzenie nadmiernych wartości, które mogą zakłócać obraz.

### 3.7 Filtr medianowy

Filtr medianowy jest skuteczną metodą odszumiania, który działa poprzez zastąpienie każdego piksela medianą wartości pikseli w jego sąsiedztwie. Jest to szczególnie efektywne w usuwaniu szumów impulsowych, ponieważ filtr ten nie jest wrażliwy na ekstremalne wartości, które mogą zafałszować wyniki.

```
fim = medfilt2(im, [k, k]);
```

Zastosowanie powyższej funkcji wykonuje tę operację, co skutkuje uzyskaniem bardziej gładkiego obrazu przy jednoczesnym zachowaniu krawędzi.



Rysunek 17: Filtr medianowy - odszumianie

Filtr medianowy skutecznie eliminuje szумy przy jednoczesnym zachowaniu konturów obiektów, co czyni go preferowaną metodą w sytuacjach, gdzie istotne jest zachowanie detali przy jednoczesnym usuwaniu zakłóceń.

### 3.8 Binaryzacja

Binaryzacja to technika przetwarzania obrazów, która polega na konwersji obrazu w odcieniach szarości na obraz binarny. W tym procesie każdy piksel jest klasyfikowany jako **czarny (0)** lub **biały (1)** w zależności od wartości intensywności oraz zastosowanego progu.

W Matlab można interaktywnie wybrać wartość progu dla binaryzacji obrazu, analizując histogram intensywności pikseli. W naszej analizie użyto dwóch progów: 0.5 i 0.6.

Dla progu 0.5, piksele o intensywności mniejszej niż 0.5 zostały ustawione na 0, a pozostałe na 1. Rezultatem tego jest obraz binarny, w którym więcej pikseli jest klasyfikowanych jako białe, co skutkuje jaśniejszym wyglądem obrazu:



Rysunek 18: Binaryzacja - próg 0.5

Z kolei przy zastosowaniu progu 0.6, więcej pikseli zostało sklasyfikowanych jako 0, co doprowadziło do uzyskania ciemniejszego obrazu binarnego:



Rysunek 19: Binaryzacja - próg 0.6

Różnice w zastosowanych progach ilustrują, jak zmiana wartości progu wpływa na finalny rezultat binaryzacji. Binaryzacja jest kluczowym krokiem w dalszym przetwarzaniu obrazów, ułatwiającym analizę konturów oraz segmentację obiektów i tła.

## 4 Laboratorium 4 - Transformacja Fouriera

Transformacja Fouriera jest używana do analizy sygnałów, umożliwiającą przekształcenie sygnału z dziedziny czasowej do dziedziny częstotliwości. W kontekście przetwarzania obrazów pozwala na analizę obrazu w kategoriach jego składowych częstotliwościowych, co jest przydatne w filtracji, kompresji oraz detekcji krawędzi obrazków.

Obraz traktowany jest jako macierz wartości intensywności pikseli. Aby przeprowadzić analizę w dziedzinie częstotliwości, wykorzystujemy dyskretną transformację Fouriera (DFT), której obliczenia w Matlab realizujemy za pomocą funkcji `fft2()`. Proces odwrócenia transformacji wykonuje funkcja `ifft2()`. Dzięki temu przekształceniu możemy oddzielnie analizować amplitudę oraz fazę sygnału:

$$z = A \cdot e^{i\phi} = A \cdot (\cos(\phi) + i \cdot \sin(\phi)), \quad (6)$$

gdzie  $A$  oznacza amplitudę, a  $\phi$  jest fazą.

Pierwszym krokiem jest wczytanie obrazu „opera.jpg”, następnie odbyła się jego konwersja do skali szarości oraz normalizacja:



Rysunek 20: Początkowy obraz „opera.jpg”

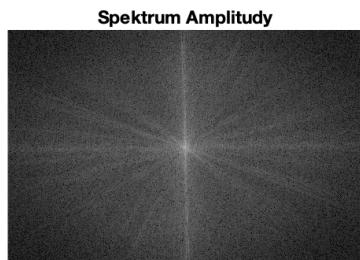


Rysunek 21: Obraz po konwersji do skali szarości i normalizacji

#### 4.1 Obliczenie transformacji Fouriera

Następnie obliczono transformację Fouriera oraz wyodrębniono amplitudę i fazę:

```
fim = fft2(im); % 2D transformacja Fouriera obrazu  
A = abs(fim); % Spektrum amplitudy  
phi = angle(fim); % Spektrum fazy
```



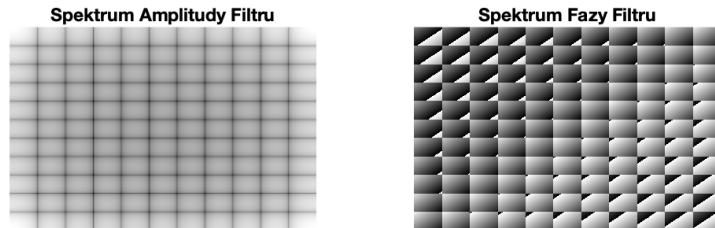
Rysunek 22: Spektrum Fazy

Amplituda wskazuje, jak mocno dana częstotliwość przyczynia się do sygnału, a faza informuje o tym, w jakim momencie sygnał osiąga swoje maksima i minima.

## 4.2 Rekonstrukcja obrazu

Rekonstrukcja obrazu z amplitudy i fazy odbyła się poprzez połączenie tych dwóch elementów. W celu filtracji zastosowano maskę o wymiarach 11x11:

```
% Rekonstrukcja obrazu z amplitudy i fazy  
z = A .* exp(1i * phi);
```



Rysunek 23: Spektrum Amplitudy i Fazy Filtru

Zastosowanie maski filtracyjnej pozwala na eliminację określonych częstotliwości, które mogą być odpowiedzialne za szумy lub niepożądane detale.

```

k = 11; % Rozmiar maski
f = ones(k) / k^2;
[h, w] = size(im);
ff = fft2(f, h, w); % transformacja Fouriera maski filtru
z = A .* fA .* exp(1i * phi); %
z = abs(ifft2(z)); % Odwrotna transformacja Fouriera,
% aby zrekonstruować obraz

```

Wynikiem zastosowania filtra był przefiltrowany obraz, który przedstawiono na poniższej ilustracji:



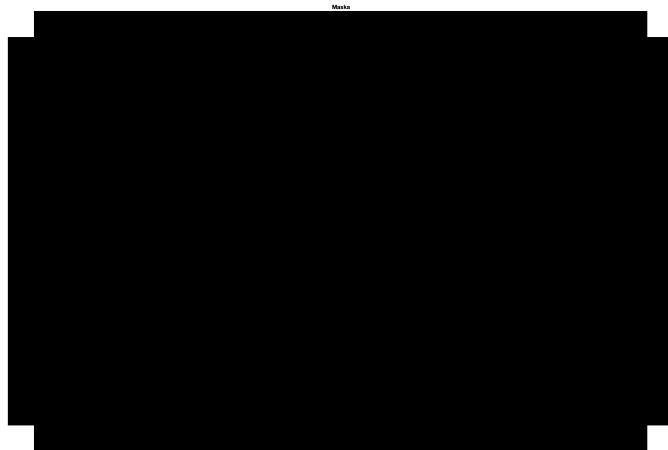
Rysunek 24: Przefiltrowany Obraz

Obraz wskazuje mniejsze szумy oraz lepsze odwzorowanie krawędzi w porównaniu do oryginału. Dzięki filtracji można poprawić jakość wizualną obrazu.

### 4.3 Kompresja obrazu

Dodatkowo, utworzono maskę binarną, którą zastosowano do spektrum amplitudy:

```
k = 50; % Rozmiar maski  
m = zeros(h, w); % Macierz zer dla maski  
m([1:k, end-k:end], [1:k, end-k:end]) = 1;
```



Rysunek 25: Maska

Ostatecznie, obraz skompresowany na podstawie zastosowanej maski:

```
z = m .* A .* exp(1i * phi);
% Odwrotna transformacja Fouriera, aby uzyskać obraz
z = abs(ifft2(z));
```



Rysunek 26: Skompresowany Obraz

Maska binarna pozwala na zachowanie tylko najistotniejszych częstotliwości, co w efekcie prowadzi do powstania skompresowanego obrazu, który

wciąż zachowuje kluczowe detale oryginalnego obrazu.

Transformacja Fouriera jest dobrym narzędziem w przetwarzaniu obrazów, umożliwiającym zarówno analizę częstotliwościową, jak i filtrowanie oraz kompresję obrazów. Przeprowadzone eksperymenty wykazały, że odpowiednia filtracja i kompresja mogą znacznie poprawić jakość i efektywność przetwarzania obrazów.

## Literatura

- [1] Wikipedia, *RGB color model*, [https://en.wikipedia.org/wiki/RGB\\_color\\_model](https://en.wikipedia.org/wiki/RGB_color_model)
- [2] Obrazki do zajęć laboratoryjnych z przedmiotu - Analiza Obrazów, [https://home.agh.edu.pl/~ggolasz/analiza\\_obrazow.html](https://home.agh.edu.pl/~ggolasz/analiza_obrazow.html)