

Analiza Obrazów

Sprawozdanie z zajęć laboratoryjnych 1-3

16. listopada 2020 r.

Autor: Tymoteusz Ciesielski

Laboratorium 1

Pierwsze faktycznie zajęcia z Analizy Obrazów były swoistym wstępem do tej dziedziny nauki. Omówiliśmy podstawy sposobu przechowywania i reprezentacji obrazów w formacie cyfrowym. Omówiliśmy model RGB przechowywania informacji o kolorach. Wykorzystaliśmy też podstawowe i najprostsze operacje możliwe do zastosowania na obrazach w pakiecie MATLAB.

Obrazem na bazie którego przeprowadzaliśmy operacje był w tym przypadku przedstawiciel dumnego gatunku żubrów europejskich (*Bison bonasus*) .



Rysunek 1 Obraz bazowy

Obraz ten został przez nas zapisany jako obiekt „im”.

Żubra poddaliśmy następnie analizie z różnych perspektyw. Na początku dokonaliśmy normalizacji wartości pikseli zawartych w grafice. Bazowo są one liczbami całkowitymi z przedziału 0-255. Przenieśliśmy je do domeny liczb rzeczywistych z przedziału 0-1, przy zachowaniu proporcji między nimi. Następnie korzystając z wbudowanych funkcji MATLABA oraz sposobu zapisu obrazu w modelu RGB, poddaliśmy żubra kilku przekształceniom.

Omawiane przez nas kolejno operacje to:

- **Filtrowanie kanałów** – odczytując tylko konkretne kanały przypisane do obrazka, możemy sprawdzić zawartość poszczególnych barw RGB w tym obrazku. Dla przykładu, odwołując się do wartości `im(:, :, 2)`, odwołujemy się wyłącznie do wartości zielonej kodowanej w obrazku. Na poniższym obrazku widać że wyraźnie na białą zaznaczone zostało otoczenie żubra – co zgodne jest z przypuszczeniem, trawa jest bowiem zielona.



Rysunek 2 Kanał z kolorem zielonym obrazu

- **Konwersja do szarości** – kolorowe obrazki można oczywiście przedstawić w szarej kolorystyce (grayscale) czyli tak zwane monochromatyczne. Każdemu pikselowi jest wtedy przypisana jedynie jedna liczba i reprezentuje ona jasność tego piksela. Jednym ze sposobów na taką operację, jest wyciągnięcie średniej z wartości RGB odpowiadających pikselom. Najprostszą taką średnią jest średnia arytmetyczna, czyli średnia ważona z równymi wagami. Powyższego dokonać można na naszym obrazie wykorzystując odpowiednią funkcję: `mean(im,3)`. Po takiej operacji, nasz ulubiony żubr wygląda następująco.



Rysunek 3 Żubr w wersji monochromatycznej

- **Inwersja kolorów** – jest to bardzo prosta operacja. Dokonuje ona niejako odwrócenia kolorystyki monochromatycznego zdjęcia – wartości białe, będą miały kolor czarny i odwrotnie. Wartości szare będą z kolei zamienione na wartości symetryczne względem połowy między zakresem białym i czarnym. Szary żubr z powyższej grafiki wyglądać będzie więc wtedy następująco.



Rysunek 4 Żubr po inwersji kolorów

- **Model YUV** – innym omawianym modelem barw był również model YUV. W przeciwieństwie do RGB, w którym to kodowane są wartości trzech kolorów, a jasność piksela zależy od sumy wartości trzech kolorów (im jest ona większa, tym większa jest jasność), w modelu YUV, osobna liczba odpowiada za wartość samej jasności (luminancję), zaś kolory kodowane są w dwóch kanałach chrominancji – U, kodującym przeskalowaną. Ciekawostką tego modelu jest fakt, że był on wykorzystywany podczas przechodzenia telewizorów czarno-białych na kolorowe. W modelu takim, przy kodowaniu obrazu w formacie YUV, zarówno użytkownicy telewizorów czarno-białych jak i kolorowych mogli cieszyć się odbiorem obrazu. Ci pierwsi korzystali jedynie z wartości Y, zaś Ci drudzy również z pozostałych dwóch wartości U i V. Konwersji z RGB do YUV możemy dokonać mnożąc odpowiednie piksele przez macierz z podanymi współczynnikami, w sposób przedstawiony poniżej.

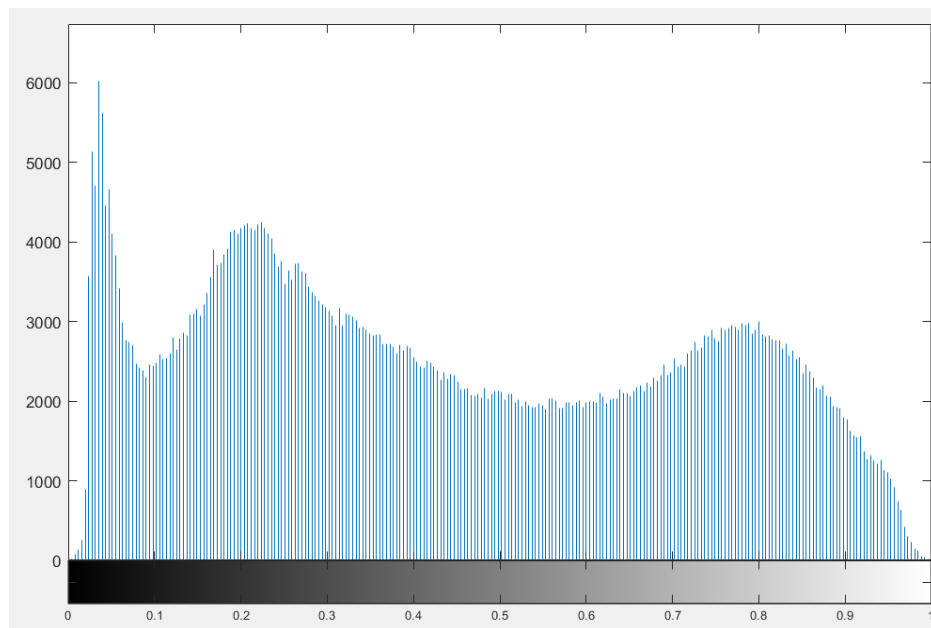
$$\begin{bmatrix} 0,299 & 0,587 & 0,114 \\ -0,147 & -0,289 & 0,437 \\ 0,615 & -0,515 & -0,100 \end{bmatrix} \cdot \begin{bmatrix} R \\ G \\ B \end{bmatrix} = \begin{bmatrix} Y \\ U \\ V \end{bmatrix}$$

Dodatkowo, z obrazu zakodowanego w modelu YUV można przejść prosto do obrazu monochromatycznego, albo rozpatrując jedynie wartość Y, bądź też ponownie wyciągając średnią z wszystkich wartości. Prawdę powiedziawszy, zastosowanie tych operacji jest tak naprawdę równoważne z zastosowaniem średniej ważonej na obrazie, z odpowiednimi wagami. Metoda ta, jest też wykorzystywana we wbudowanej funkcji `rgb2gray(im)`. Wynik obu jest taki sam i został przedstawiony poniżej.



Rysunek 5 Żubr monochromatyczny YUV

Ważnym narzędziem wykorzystywanym przy analizie obrazów jest też histogram. Dla obrazów monochromatycznych rozpatrywane wartości to odcień – po znormalizowaniu, im bliżej zera tym bliższy czarnemu, im bliżej 1 tym bliższy białemu. Dla powyższego obrazu, histogram przedstawiono poniżej.



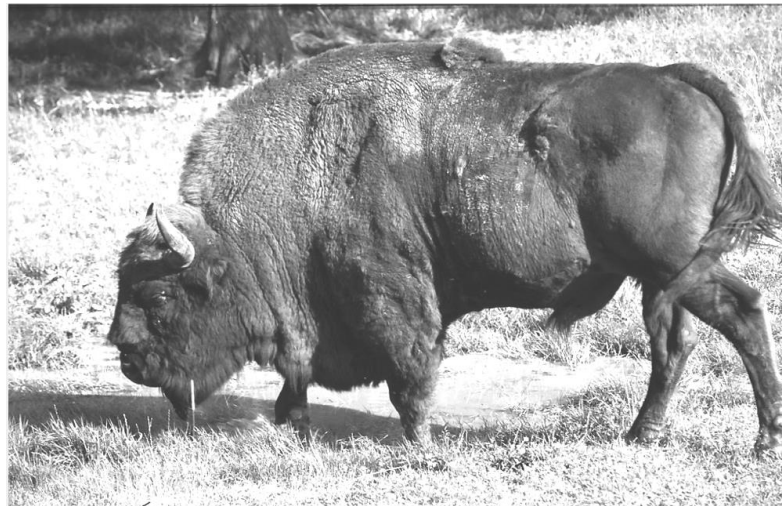
Rysunek 6 Histogram rozkładu jasności monochromatycznego żubra

Analizując trzy wyraźne maksima, możemy wyodrębnić je jako: pierwsze maksimum to prawie czarne rejony ciała żubra i fragmenty lasu. Drugie maksimum, to ciało żubra, trzecie maksimum to trawa.

- **Modyfikacja jasności obrazu** – operację tę, również możemy zastosować na kilka sposobów. Jednym z prostszych jest dodanie bądź odjęcie pewnej stałej do wartości wszystkich pikseli. Dodając, zwiększamy jasność. Odejmując, zmniejszamy ją. Pojawiają się jednak potencjalne problemy – co z wyjściem poza skalę wartości dla pikseli? Rozwiązać możemy to na kilka sposobów m.in.:

- Piksele które uzyskały wartość mniejszą od 0, ściągane są do wartości 0. Z kolei piksele które osiągnęły wartość większą od 1 uzyskują wartość 1.
- Dokonujemy przeskalowania wartości pikseli w uzyskanym obrazie. Nowa największa wartość, która wyszła poza skalę, stanowi wartość maksymalną, zaś nowa najmniejsza wartość stanowi wartość minimalną. Wszystkie wartości pikseli zostają odpowiednio przeskalowane w oparciu o dwie wspomniane wartości

Przy zastosowaniu tej pierwszej metody, i po dodaniu do pikseli w znormalizowanym obrazie wartości 0.2, otrzymujemy następujący rezultat:



Rysunek 7 Żubr po dodaniu do pikseli wartości 0.2

Analogicznie, wartości pikseli w obrazie możemy też pomnożyć. Przykładowy rezultat, dla pomnożenia wartości przez 0.5 widać poniżej (oczywistym jest że obraz ten będzie ciemniejszy).



Rysunek 8 Żubr po pomnożeniu wartości pikseli przez 0.5

Na obrazie, w zależności od potrzeby, zastosować możemy też bardziej skomplikowane złożenia tych operacji.

- **Korekcja gamma** – najczęściej stosowana korekcja jasności w przypadku urządzeń elektronicznych takich jak telefony czy tablety. Wybierana jest wartość współczynnika gamma, a następnie piksele wartości są obliczane według wzoru:
$$\text{gamma_aim} = \text{gim}^{(1/\text{gamma})}$$

Dla gamma będącego zazwyczaj wartością od ¼ do 4. Dla znormalizowanego obrazu, im większy parametr gamma tym jaśniejszy obraz. Poniżej żubr dla gamma = 2.



Rysunek 9 Żubr po zastosowaniu korekcji gamma dla gamma = 2

Pod koniec zajęć laboratoryjnych zdałem sobie sprawę, że w nawet tak z pozoru błahym zagadnieniu, jak konwersja obrazu do odcieni szarości i zmiana jasności, kryje się ukryta głębia i prawdziwe spektrum różnego rodzaju mechanizmów. Nawet nie podejrzewałem jaka rewolucja czekać będzie na mnie podczas kolejnych zajęć.

Laboratorium 2

Podczas tych zajęć laboratoryjnych zajęliśmy się szerokim zagadnieniem jakim są filtry. Rozpoczęliśmy od przyjęcia wspólnych definicji bazowych jakimi są sąsiedztwo von Neumana i sąsiedztwo Moore'a.

Część praktyczną zainaugurowaliśmy konwersją naszego starego przyjaciela żubra do obrazu monochromatycznego.

Zdefiniowaliśmy nasze podstawowe filtry, będące macierzami wag, o sumie elementów wynoszącej 1 i rozpoczęliśmy od skorzystania z funkcji `imfilter()`. Stosowaliśmy kolejno filtry:

- **Filtr dolnoprzepustowy** – filtr ten jest zwany również filtrem odszumiającym. Jego zadaniem jest rozmycie krawędzi, zmniejszenie udziału szczegółów w obrazie czyli właśnie pozbycie się szumu. Przykładowy filtr o wymiarach 7x7 z wagami każdego elementu 1/7 zastosowany na naszym żubrze daje następujący efekt:



Rysunek 10 Żubr przepuszczony przez filtr dolnoprzepustowy

Rozmycie jest ewidentnie widoczne.

- **Filtr górnoprzepustowy** – filtr ten zwany jest filtrem wyostrzającym. Jego zadaniem jest podniesienie kontrastu oraz wyróżnienie szczegółów na obrazie. Doprowadza on też do zwiększenia szumu na obrazie. Poniżej skorzystaliśmy z filtra o rozmiarze 3x3, elemencie centralnym o wartości 9 a pozostałych o wartości -1.



Rysunek 11 Żubr przepuszczony przez filtr górnoprzepustowy

- **Filtr medianowy** – stanowi odmienne podejście do stosowanej do tej pory średniej ważonej. Działa on podobnie do filtra wykorzystującego średnie, rozmywa obraz, z drugiej strony po jego zastosowaniu krawędzie stają się jasno oddzielone przez co łatwiej jest je wykryć. Korzystaliśmy tutaj z funkcji `medfilt2(gim);`



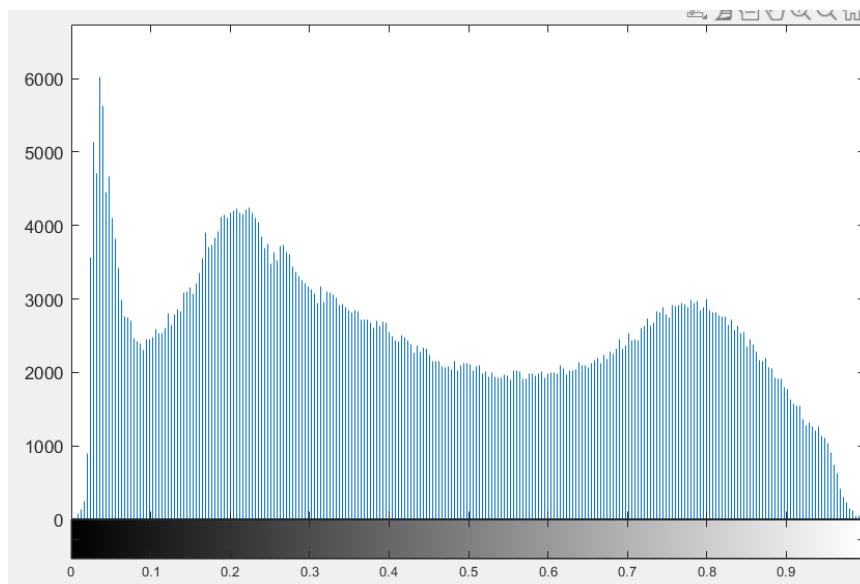
Rysunek 12 Żubr przepuszczony przez filtr medianowy

Następnie zajęliśmy się stosunkowo mało skomplikowaną operacją jaką jest binaryzacja obrazu czyli sprowadzenie wszystkich pikseli do wartości 0 lub 1 – innymi słowy, sprawienie że nasz obraz jest perfekcyjnie czarno-biały. Ustaliliśmy arbitralnie granicę odcięcia między tymi dwoma wartościami na poziomie 0.6 a następnie dokonaliśmy binaryzacji korzystając z funkcji `imbinarize()` co podziwiać można poniżej. Wartości pikseli zostały również zanegowane, ponieważ zazwyczaj rozważany przez nas obiekt chcemy traktować jako biały, z czarnym tłem.



Rysunek 13 Żubr zbinaryzowany dla wartości progowej = 0.6

Widzimy że rezultat nie jest idealny – żubr nie jest bardzo dobrze odróżnialny od tła. Świadomi działania metody z jakiej korzystamy, powinniśmy zdać sobie oczywiście sprawę że dobór wartości progowej stanowi tutaj ważny czynnik. Powinniśmy go dobrać tak odpowiednio aby ciało żubra znajdowało się w jednej kategorii (białe) zaś otoczenie w drugiej (czarnej). Przydatny w tej analizie jest rozważany już przez nas wcześniej histogram.



Rysunek 14 Histogram rozkładu jasności monochromatycznego żubra

Warto zwrócić też uwagę że wartość parametru powinniśmy dobierać z minimum lokalnego raczej niż z maksimum – jest to intuicyjne, chcemy żeby wiele bardzo podobnych elementów należało do jednej kategorii, zamiast stawiać między nimi granicę. Analizując histogram nową wartość progu dobrać możemy jako 0.4. Sprawdźmy jak poradziłyśmy sobie tym razem.



Rysunek 15 Zbinaryzowany żubr dla wartości progowej = 0.4

Widać że kontury żubra są dużo bardziej wyraźne, większa część tła została odcięta – sukces!

Kolejne operacje są bardzo istotne i interesujące pod kątem analizy obrazów.

- **Dylatacja** – wykorzystywana funkcja to `imdilate()`. Inna jej nazwa to akumulacja. Precyzyjnie, sprawia ona, że dla każdego piksela, jeśli w jego przyjętym sąsiedztwie znajduje się piksel koloru białego, to piksel centralny również przyjmuje wartość białą. Działanie można nieformalnie określić jako nakładanie się na krawędzie obiektu nowych warstw. Im większe sąsiedztwo przyjmiemy, tym bardziej „pogrubimy” nasz obiekt. Poniżej efekt dylatacji dla elementu strukturalnego przyjętego jako macierz 3x3 i zbinaryzowanego obrazu z poprzedniego podpunktu.



Rysunek 16 Żubr poddany dylatacji

- **Erozja** – jest to operacja odwrotna do dylatacji, realizowana za pomocą funkcji `imerode()`. Zmienia ona piksele na czarne, jeśli w ich sąsiedztwie znajduje się choć jeden czarny piksel. Nieformalnie, dokonuje ona „odchudzenia” obrazu i pogłębienia tak zwanych wąwozów. Poniżej efekt działania erozji dla elementu strukturalnego macierzy 3x3 i wspomnianego zbinaryzowanego obrazu.



Rysunek 17 Żubr poddany erozji

Faktycznie, ewidentnie wydaje się jakby nasz żubr przeszedł na dietę.

- **Zamknięcie** – operacja ta jest tak naprawdę złożeniem dwóch poprzednich – kolejność ma jednak w tym przypadku znaczenie. Najpierw dokonujemy tutaj dylatacji a następnie erozji. Wyjściowy obraz nie jest jednak tożsamy z wejściowym. Po tych operacjach małe czarne obszary na obiekcie zostają zmienione na białe lub „zamknięte”. Korzystamy tutaj z tych dwóch operacji we wspomnianej kolejności lub też z wbudowanej funkcji `imclose()`. Efekt działania dla elementu strukturalnego macierzy 5x5 poniżej.



Rysunek 18 Żubr poddany operacji zamknięcia

- **Otwarcie** – analogicznie, operacja ta jest złożeniem kolejno erozji i dylatacji. Powoduje ona „otwarcie obszarów” blisko krawędzi obrazu, lub też połączenie się dziur wewnątrz obiektu. korzystamy tu z funkcji `imopen()` a efekt działania dla elementu strukturalnego macierzy 5x5 obserwujemy poniżej.



Rysunek 19 Żubr poddany operacji otwarcia

Laboratorium 3

Na trzecich zajęciach laboratoryjnych, mieliśmy szansę odbyć małą podróż sentymentalną i cofnąć się pamięcią do sielskich czasów zajęć z analizy matematycznej i Transformaty Fouriera. Stosowaliśmy ją w kontekście modyfikacji obrazów oraz ich filtracji. Korzystaliśmy także z algorytmu tak zwanej Szybkiej Transformaty Fouriera – funkcji `fftshift()` oraz `fft2()`.

Zmieniliśmy również rozważany obraz na coś bardziej egzotycznego – jeden z 21 cudów świata nowożytnego, arcydzieło architektury – Operę w Sydney, którą w oryginalnej wersji podziwiać możemy poniżej.



Rysunek 20 Opera bazow

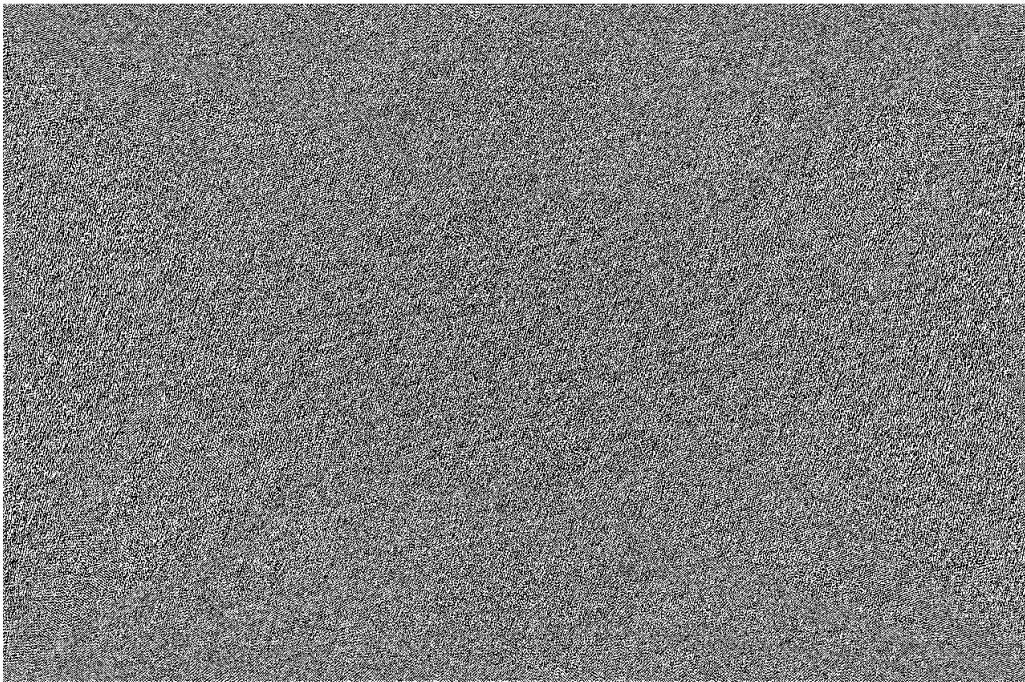
a

Z kolei w wersji monochromatycznej również nie odstaje wiele od oryginału.



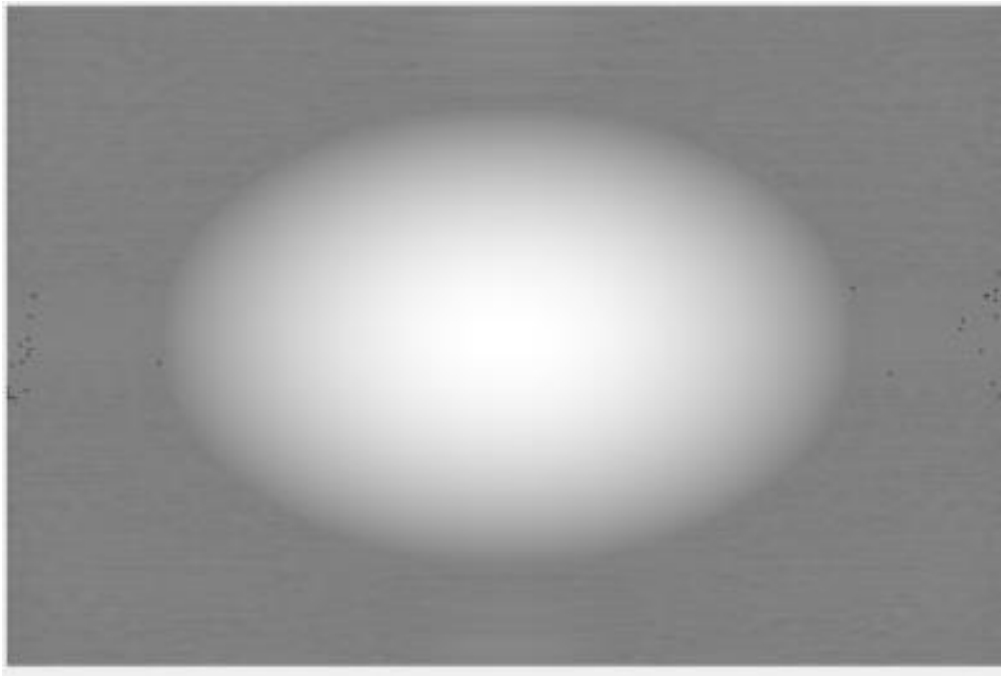
Rysunek 21 Opera monochromatyczna

Po zastosowaniu Szybkiej Transformaty Fouriera.



Rysunek 22 Opera w domenie falowej

Nie otrzymujemy jednak niczego spektakularnego – przeszliśmy do domeny falowej, a konkretniej jest to widmo fazowe tego obrazu. Z tak zakodowanego obrazu otrzymać jednak możemy kąt oraz właśnie fazę, a następnie maksymalną amplitudę. Rozważyć możemy także widmo amplitudowe.



Rysunek 23 Widmo amplitudowe obrazu

Podczas zajęć, na wspomnianym obrazie wykorzystaliśmy także filtr Gaussa.



Rysunek 24 Opera po zastosowaniu filtru z maską Gaussa

Pod koniec obserwujemy złożenie kilku różnych operacji: zmiany fazy obrazu w domenie falowej, inwersji nałożenia masek oraz kompresji. Skomplikowanie niniejszych obecnie przekracza moje zdolności kognitywistyczne ale efekt jest zdecydowanie zadowalający.



Rysunek 25 Opera po zmianie fazy, inwersji, nałożeniu maski i kompresji