



AKADEMIA GÓRNICZO-HUTNICZA

im. Stanisława Staszica w Krakowie

**WYDZIAŁ INŻYNIERII
MECHANICZNEJ I ROBOTYKI**

Praca dyplomowa inżynierska

Grzegorz Tworek

Imię i nazwisko

Mechanika i Budowa Maszyn

Kierunek studiów

**„Numeryczne rozwiązanie problemu
odwrotnego - rekonstrukcja obrazu”**

Temat pracy dyplomowej

dr inż. Katarzyna Suder-Dębska

Promotor pracy

.....

Ocena

Kraków, rok 2018/2019

Kraków, dn.....

Imię i nazwisko: Grzegorz Tworek
Nr albumu: 285726
Kierunek studiów: **Mechanika i Budowa Maszyn**
Profil dyplomowania: Informatyka w Inżynierii Mechanicznej

OŚWIADCZENIE

Świadomy/a odpowiedzialności karnej za poświadczanie nieprawdy oświadczam, że niniejszą inżynierską pracę dyplomową wykonałem/łam osobiście i samodzielnie oraz nie korzystałem/łam ze źródeł innych niż wymienione w pracy.

Jednocześnie oświadczam, że dokumentacja oraz praca nie narusza praw autorskich

w rozumieniu ustawy z dnia 4 lutego 1994 roku o prawie autorskim i prawach pokrewnych (Dz. U. z 2006 r. Nr 90 poz. 631 z późniejszymi zmianami) oraz dóbr osobistych chronionych prawem cywilnym. Nie zawiera ona również danych i informacji, które uzyskałem/łam w sposób niedozwolony. Wersja dokumentacji dołączona przeze mnie na nośniku elektronicznym jest w pełni zgodna z wydrukiem przedstawionym do recenzji.

Zaświadczam także, że niniejsza inżynierska praca dyplomowa nie była wcześniej podstawą żadnej innej urzędowej procedury związanej z nadawaniem dyplomów wyższej uczelni lub tytułów zawodowych.

.....
podpis dyplomanta

Kraków,

Imię i nazwisko: Grzegorz Tworek

Adres korespondencyjny: Nowy Kamień 98, 27-600 Sandomierz

Temat pracy dyplomowej inżynierskiej: **Numeryczne rozwiązanie problemu**

Rok ukończenia: 2019

Nr albumu: 285726

Kierunek studiów: **Mechanika i Budowa Maszyn**

Profil dyplomowania: Informatyka w Inżynierii Mechanicznej

OŚWIADCZENIE

Niniejszym oświadczam, że zachowując moje prawa autorskie, udzielam Akademii Górniczo-Hutniczej im. S. Staszica w Krakowie nieograniczonej w czasie nieodpłatnej licencji niewyłącznej do korzystania z przedstawionej dokumentacji inżynierskiej pracy dyplomowej, w zakresie publicznego udostępniania i rozpowszechniania w wersji drukowanej i elektronicznej¹.

Publikacja ta może nastąpić po ewentualnym zgłoszeniu do ochrony prawnej wynalazków, wzorów użytkowych, wzorów przemysłowych będących wynikiem pracy inżynierskiej².

Kraków,
data *podpis dyplomanta*

¹ Na podstawie Ustawy z dnia 27 lipca 2005 r. Prawo o szkolnictwie wyższym (Dz.U. 2005 nr 164 poz. 1365) Art. 239. oraz Ustawy z dnia 4 lutego 1994 r. o prawie autorskim i prawach pokrewnych (Dz.U. z 2000 r. Nr 80, poz. 904, z późn. zm.) Art. 15a. "Uczelni w rozumieniu przepisów o szkolnictwie wyższym przysługuje pierwszeństwo w opublikowaniu pracy dyplomowej studenta. Jeżeli uczelnia nie opublikowała pracy dyplomowej w ciągu 6 miesięcy od jej obrony, student, który ją przygotował, może ją opublikować, chyba że praca dyplomowa jest częścią utworu zbiorowego."

² Ustawa z dnia 30 czerwca 2000r. – Prawo własności przemysłowej (Dz.U. z 2003r. Nr 119, poz. 1117 z późniejszymi zmianami) a także rozporządzenie Prezesa Rady Ministrów z dnia 17 września 2001r. w sprawie dokonywania i rozpatrywania zgłoszeń wynalazków i wzorów użytkowych (Dz.U. nr 102 poz. 1119 oraz z 2005r. Nr 109, poz. 910).

Kraków, dnia

**AKADEMIA GÓRNICZO-HUTNICZA
WYDZIAŁ INŻYNIERII MECHANICZNEJ I ROBOTYKI**

TEMATYKA PRACY DYPLOMOWEJ INŻYNIERSKIEJ
dla studenta IV roku studiów stacjonarnych

.....
imię i nazwisko studenta

TEMAT PRACY DYPLOMOWEJ INŻYNIERSKIEJ:

Numeryczne rozwiązanie problemu odwrotnego - rekonstrukcja obrazu

Promotor pracy: [dr inż. Katarzyna Suder-Dębska]

Recenzent pracy: [dr hab. inż. Marian Banaś]

.....
Podpis dziekana:

PLAN PRACY DYPLOMOWEJ (Należy ustalić z Promotorem, np.):

1. Omówienie tematu pracy i sposobu realizacji z promotorem.
2. Zebranie i opracowanie literatury dotyczącej tematu pracy.
3. Zebranie i opracowanie wyników badań.
4. Analiza wyników badań, ich omówienie i zatwierdzenie przez promotora.
5. Opracowanie redakcyjne.

Kraków,
data *podpis dyplomanta*

TERMIN ODDANIA DO DZIEKANATU: 20..... r.

.....
podpis promotora

Akademia Górniczo-Hutnicza im. Stanisława Staszica
Wydział Inżynierii Mechanicznej i Robotyki

Kraków,

Kierunek: Mechanika i Budowa Maszyn

Profil dyplomowania: Informatyka w Inżynierii Mechanicznej

Grzegorz Tworek

Praca dyplomowa inżynierska

Numeryczne rozwiązanie problemu odwrotnego - rekonstrukcja obrazu

Opiekun: dr inż. Katarzyna Suder-Dębska

STRESZCZENIE

Celem pracy było rozwiązanie problemu odwrotnego poprzez przeprowadzenie rekonstrukcji zaszumionego obrazu z wykorzystaniem metod zaimplementowanych w środowisku Matlab. Rezultaty zastosowanych metod zostały następnie porównane w oparciu o histogramy odtworzonych obrazów. Rekonstrukcje wykonano dwoma opisanymi metodami tj. algorytm Lucy-Richardsona i filtr Wienera.

Na przykładowy obraz został najpierw nałożony szum, a następnie wykonano rekonstrukcje wyżej wymienionymi metodami. Działanie metod usprawniano poprzez dodanie parametrów z odpowiednimi wartościami w celu uzyskania jak najlepszej jakości odtworzenia. Odtworzone obrazy przedstawiono razem z histogramami, które opisano i porównano pod względem efektywności.

AGH University of Science and Technology
Faculty of Mechanical Engineering and Robotics

Kraków, the.....

Field of Study: Mechanical engineering

Specialisations: Informatics in Mechanical Engineering

Grzegorz Tworek

Engineer Diploma Thesis

Numerical solution of the inverse problem - reconstruction of the photo

Supervisor: dr inż. Katarzyna Suder-Dębska

SUMMARY

The main objective of this thesis was to solve the inverse problem by reconstructing noisy image with methods implemented in the Matlab environment. Results of the used methods were compared with the usage of collected data that are presented in histograms. Reconstruction was made using two methods, the Lucy-Richardson algorithm and the Wiener filter.

First, noise was added to the sample picture and after that a reconstruction was made with the earlier mentioned methods. These methods were improved by adding parameters with appropriate values to obtain the best results. The reconstructed images were presented with histograms and compared in terms of efficiency.

Spis treści

1. Wstęp.....	3
1.1 Cel pracy.....	3
1.2 Zakres pracy.....	3
2. Problem odwrotny - charakterystyka.....	4
2.1 Opis zagadnienia.....	4
2.2 Problem odwrotny w rekonstrukcji obrazów.....	4
3. Reprezentacja obrazów.....	6
3.1 Czym jest obraz?.....	6
3.2 Układ obrazu (layout).....	6
3.3 Intensywność obrazu (kolor).....	7
3.4 Format obrazu.....	7
3.5 Rozdzielczość i kwantyzacja.....	8
3.6 Kompresja obrazów.....	9
4. Formowanie obrazów.....	10
4.1 Jak powstaje obraz?.....	10
4.2 Filtracja cyfrowa.....	10
4.3 Konwolucja.....	12
4.4 Kwantyzacja.....	14
4.5 Szum w obrazach.....	15
5. Histogramy.....	16
5.1 Wyrównywanie histogramów.....	17
5.2 Histogramy obrazów kolorowych.....	18
6. Filtrowanie szumów.....	19
6.1 Filtr uśredniający.....	19
6.2 Filtr medianowy.....	19
6.3 Filtr Gaussowski.....	20
7. Rekonstrukcja obrazów.....	21
8. Zastosowanie algorytmu Lucy-Richardsona do rekonstrukcji obrazu.....	23
8.1 Implementacja funkcji dekonwolucji algorytmem Lucy-Richardsona.....	24
8.2 Rekonstrukcja obrazu algorytmem Lucy-Richardsona.....	26
8.3 Wnioski.....	35
9. Zastosowanie filtru Wienera do rekonstrukcji obrazu.....	36
9.1 Implementacja filtru Wienera.....	37
9.2 Rekonstrukcja obrazu filtrem Wienera.....	38

9.3 Wnioski.....	46
10. Podsumowanie.....	47
Bibliografia.....	48
Załączniki:	49

1. Wstęp

Rekonstrukcja obrazów czy też ich przywracanie jest dość starym problemem w dziedzinie przetwarzania obrazów. Zagadnienie przyciąga uwagę wielu środowisk naukowych. Ze względu na swoją użyteczność rekonstrukcja obrazów znajduje zastosowanie w wielu rzeczywistych problemach. Kwestie takie jak identyfikacja ludzi, nadzorowanie przemysłu, oprogramowanie robotów czy analiza obrazów astronomicznych można efektywnie opracowywać wykorzystując algorytmy odtwarzania obrazów. Ponadto przywracanie obrazu jest jednym z przykładów szerokiej gamy zastosowań problemów odwrotnych. Zagadnienia odwrotne pojawiają się we wszystkich rodzajach problemów naukowych.

1.1 Cel pracy

Celem pracy jest rekonstrukcja zaszumionego obrazu z wykorzystaniem metod rekonstrukcji zaimplementowanych w środowisku Matlab. Następnie porównanie ich pod względem efektywności w oparciu o uzyskane histogramy.

1.2 Zakres pracy

W części teoretycznej pracy przedstawiono zagadnienie problemu odwrotnego, opisano reprezentacje obrazów. Przedstawiono formowanie obrazów z podstawowymi przekształceniami. Następnie zdefiniowano pojęcie histogramu i zaprezentowano kilka jego przykładów. Omówiono filtrację linową oraz działanie wybranych filtrów usuwających szum.

W części projektowej scharakteryzowano zastosowane metody tj. filtr Wienera i algorytm Lucy-Richardsona, dokonano ich implementacji w środowisku Matlab oraz opisano efekty ich działania na obraz ze względu na użyte parametry. Na zakończenie przedstawiono wyniki rekonstrukcji na uzyskanych histogramach porównano rezultaty i wyciągnięto wnioski.

2. Problem odwrotny - charakterystyka

Przez ostatnie dziesięciolecia metody numeryczne z wykorzystaniem iteracyjnej rekonstrukcji stały się dominującym podejściem do rozwiązywania problemów odwrotnych w kategorii cyfrowego przetwarzania obrazów. Zagadnienia takie jak odszumianie, dekonwolucja i interpolacja, mogą być teraz efektywnie wykonywane. Dzięki pojawieniu się sensorów skompresowanych oraz powiązanych z nimi regularyzatorów, powstały solidne i praktyczne algorytmy pozwalające uzyskać świetną jakość obrazu przy niedużej złożoności obliczeń. Wymienione wyżej postępy miały ogromne znaczenie zwłaszcza w dziedzinie obrazowania biomedycznego, np. w obrazowaniu rezonansem magnetycznym czy tomografii komputerowej. Metoda iteracyjnej rekonstrukcji z regularyzacją jest jednym ze sposobów na poradzenie sobie z powyższymi problemami na etapie programowania, tzn. bez konieczności stosowania nowych skanerów.

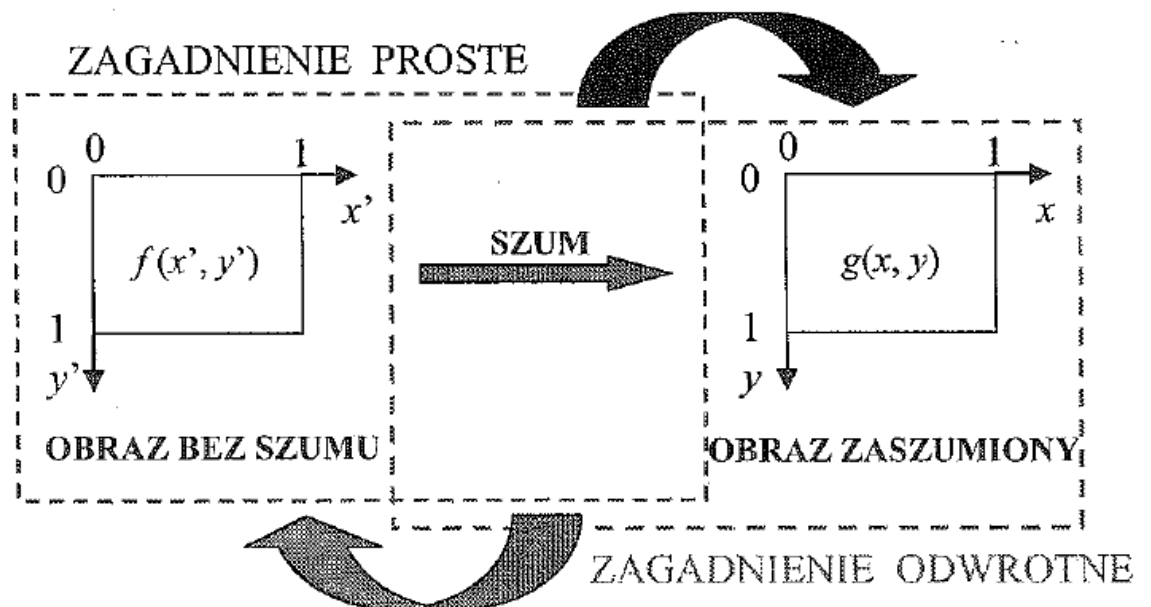
2.1 Opis zagadnienia

Aby dobrze zaprezentować działanie problemów odwrotnych zostanie przytoczony przykład w kontekście propagacji fali. Mając źródło i warunki brzegowe można obliczyć strukturę fal i ich kształt praktycznie w każdej lokalizacji w czasoprzestrzeni. Takie zagadnienie nazywa się prostym lub problemem prostym. Problem odwrotny polega na wykorzystaniu obserwacji fali, wzdłuż jednej lub kilku powierzchni, do wyznaczenia źródła fali. Może się też zdarzyć, że otrzymane dane są niekompletne wtedy problem odwrotny jest źle zdefiniowany. Oznacza to, że istnieje kilka prawdopodobnych źródeł, które spowodowały daną obserwację i nie można ustalić, które jest prawdziwe. Problemy odwrotne są dziedziną nauki, której zadaniem jest określić, kiedy problem jest źle zdefiniowany i do jakiego stopnia, a następnie wydobyć z niego maksimum informacji.

2.2 Problem odwrotny w rekonstrukcji obrazów

Dla przypadku ciągłego niech $H: L^2(\mathbb{R}^d) \rightarrow L^2(\Omega)$ będzie operatorem liniowym, gdzie $L^2(X) = \{f: X \rightarrow \mathbb{C} \mid \int_X |f(x)|^2 dx < +\infty\}$. Wtedy H jest operatorem liniowym odwzorowującym od funkcji 1D (jednowymiarowych) o wartościach zespolonych z funkcją kwantyfikacji kwadratowej (obrazy) do funkcji zespolonych z kwantyfikacją kwadratową w innej przestrzeni, Ω (pomiar). Przestrzeń $\Omega \subseteq \mathbb{R}^d$ pozostaje uogólniona, aby uwzględnić operatory gdzie pomiary nie przyjmują naturalnie

postaci obrazu. Dla przykładu pomiary mogą być próbkami punktowymi obrazu w znanych miejscach lub całkami liniowymi obrazu indeksowanego, przez ich orientację. Niech H^* będzie operatorem, funkcją łącznikową zdefiniowaną w taki sposób, że $\langle f, H^* g \rangle = \langle Hf, g \rangle$. Wtedy zgodnie z zagadnieniem odwrotnym interesujący jest taki dobór parametrów f i g , aby rzeczywiście mogło się dać postawić znak równości w powyższym równaniu. Przykładowy schemat rekonstrukcji obrazu jest przedstawiony na rys. 1.



Rys. 1 Rekonstrukcja obrazu na podstawie zarejestrowanego obrazu zaszumionego [1]

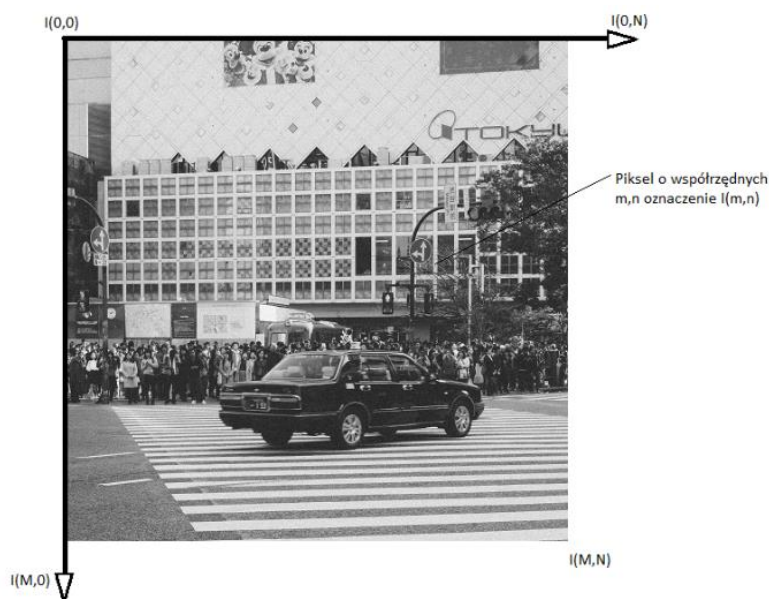
3. Reprezentacja obrazów

3.1 Czym jest obraz?

Obraz cyfrowy można definiować jako dyskretną reprezentację danych posiadających przestrzeń (układ) i intensywność (kolor) informacji. Do opisu obrazu można posługiwać się też takimi terminami jak mapa bitowa czy zbiór pikseli. W niektórych przypadkach obraz można potraktować również jako wielowymiarowy sygnał.

3.2 Układ obrazu (layout)

Dwuwymiarowy (2-D) dyskretny cyfrowy obraz $I(m,n)$ reprezentuje odpowiedź odpowiednich sensorów w szeregu stałych pozycji ($m=1,2,\dots,M$; $n=1,2,\dots,N$) w dwuwymiarowym układzie kartezjańskim. Dwuwymiarowy ciągły sygnał przestrzenny wynika z procesu próbkowania. Próbkowanie nosi też miano dyskretyzacji. Dyskretyzacja występuje w sposób naturalny w niektórych rodzajach czujników np. CCD, gdzie dokonuje uśredniania lokalnego sygnału ciągłego do odbiornika. Wskaźniki m i n oznaczają wiersze i kolumny obrazu, a zatem wynika z tego, że poszczególne elementy czy piksele są określane w układzie 2-D. Odpowiedź piksela $I(m,n)$, znajdującego się w m rzędzie i n kolumnie liczona jest od górnego lewego rogu obrazu zgodnie z konwencją Matlaba (Rys. 2).



Rys 2. Rozkład współrzędnych pikseli na dwuwymiarowym układzie kartezjańskim [8]

Pojedyncze wartości pikseli odpowiadają pewnym fizycznym wartościom w większości dwuwymiarowych obrazów rzeczywistych (np. intensywność optyczna uzyskiwana na obrazie z kamery). Obecnie można swobodnie interpretować obrazy w abstrakcyjnych przestrzeniach, gdzie współrzędne nie odpowiadają przestrzeni fizycznej, a także interpretować obrazy o trzech lub więcej wymiarach. Przykładowo obrazowanie medyczne uwzględnia pełną trójwymiarową (3-D) rekonstrukcję narządów.

3.3 Intensywność obrazu (kolor)

Obrazy zawierają jeden lub więcej kolorów, które definiowane są poprzez intensywność lub barwę w odpowiednim pikselu. W najprostszym przypadku jedna lokalizacja piksela ma tylko jedną wartość liczbową określającą poziom sygnału w danym punkcie obrazu. Konwersja do obrazu rzeczywistego odbywa się poprzez kolorową mapę. Kolorowa mapa ma za zadanie przypisać konkretny odcień koloru do każdego poziomu numerycznego, aby nadać mu wizualny charakter. Najbardziej popularną kolorową mapą jest skala szarości, która przypisuje odcieniom szarości wartości od 0 do 255, gdzie 0 oznacza maksymalną czerń, a 255 maksymalną jasność. Skala szarości jest szczególnie dobrze dostosowana do obrazów o dużej intensywności, czyli takich gdzie intensywność sygnału jest określona przez pojedynczą wartość w regionie.

Oprócz skali szarości gdzie pojawia się dokładnie jedna wartość liczbowa dla każdej lokalizacji pikseli istnieją także obrazy o prawdziwym kolorze gdzie obrazy są ukazywane w pełnym spektrum kolorów. Do reprezentacji kolorów służy wektor potrójny (R,G,B), w którym kolor jest przedstawiony jako liniowa kombinacja trzech kolorów bazowych (czerwony, zielony, niebieski), albo jako obraz składający się z trzech płaszczyzn 2-D.

3.4 Format obrazu

Pod względem matematycznym każda dwuwymiarowa tablica liczb jest obrazem. Jednakże w prawdziwym świecie konieczne jest skuteczne wyświetlanie, przechowywanie i przesyłanie obrazów, a także rozpoznawanie odpowiednich danych liczbowych jako takich, które odpowiadają tym obrazom. Z tego względu powstały standardowe formaty obrazów cyfrowych. Niektóre z najbardziej popularnych formatów przedstawiono w tabeli 1.

Tabela 1. Podstawowe formaty obrazów i ich właściwości [3]

Akronim	Nazwa	Właściwości
GIF	Graphics interchange format	Ograniczony do 256 kolorów, kompresja bezstratna
JPEG	Joint Photographic Experts Group	Najczęściej stosowany format, kompresja stratna, występują też warianty z kompresją bezstratną
BMP	Bit map picture	Podstawowy format, głównie stosowana kompresja bezstratna, występuje kompresja stratna
PNG	Portable network graphics	Format z kompresją bezstratną, stworzony by zastąpić GIF
TIF/TIFF	Tagged image (file) format	Bardzo elastyczny i szczegółowy format z wariantami skompresowanymi i nieskompresowanymi

3.5 Rozdzielczość i kwantyzacja

Rozmiar siatki dwuwymiarowego obrazu jest określony przez rozdzielczość i kwantyzację kolorów. Rozdzielczość można określić w trzech różnych wielkościach, jako:

- Rozdzielczość przestrzenną, czyli wymiary kolumn i wierszy, co stanowi liczbę pikseli użytych do pokrycia przestrzeni wizualnej uchwyconej na obrazie. Odnosi się to do próbkowania sygnału i rozmiaru cyfrowego podawanego w stosunku liczby kolumn do liczby wierszy.

- Rozdzielczość czasową, która jest stosowana w systemach ciągłego przechwytywania obrazu, takich jak format wideo. Określa się w ten sposób liczbę obrazów w jednostce czasu (powszechnie stosowane klatki na sekundę).
- Rozdzielczość bitową, która określa liczbę możliwych intensywności/kolorów jakie może osiągnąć piksel i odnosi się do kwantyzacji. Przykładowo obraz binarny ma dwa kolory czarny i biały, obraz w skali szarości ma ich 256, a dla obrazu kolorowego zależy to od stosowanego zakresu. Rozdzielczość bitowa zgodnie z nazwą jest określana przez liczbę bitów potrzebną do przechowania danych na danym poziomie kwantyzacji: dla binarnych 2 bity, dla skali szarości 8 bitów, dla kolorowych 24 bity.

3.6 Kompresja obrazów

Kolejnym z ważnych czynników przy przechowywaniu obrazów jest jego kompresja. Kompresja pozwala na to, że obraz zajmuje mniej miejsca na dysku oraz można go przenosić w krótszym czasie.

Kompresja stratna polega na zmniejszeniu ilości bitów wykorzystywanych do opisu informacji na obrazie. Algorytm kompresji odrzuca najmniej istotne dane pozostawiając te bardziej wartościowe. Zdarzają się przypadki, że niektóre obrazy mają nadmiar informacji i wtedy stosowanie tego rodzaju kompresji wpływa korzystnie. Istnieją dwie formy informacji, którą można usunąć. Może to być zmniejszenie ilości szczegółów, albo zmniejszenie liczby kolorów/poziomów szarości. Niektóre z formatów przedstawionych w tabeli 1 wykorzystują kompresję stratną. Kompresja stratna jest powszechnie stosowana w formatach wideo ze względu na dużą ilość danych źródłowych powiązanych z liczbą klatek na sekundę. Taki sposób utraty informacji jest pewną formą szumu i może ograniczyć możliwości późniejszej poprawy i analizy obrazu. Gdy w Matlabie wczytuje się obraz w formacie JPEG, który wykorzystuje kompresję stratną należy pamiętać, że nie będzie on przechowywany jako dokładny.

Kompresja bezstratna z kolei zmniejsza objętość obrazu w pamięci masowej co może się przyczynić niekiedy do dużej utraty szczegółów względem oryginalnego obrazu.

4. Formowanie obrazów

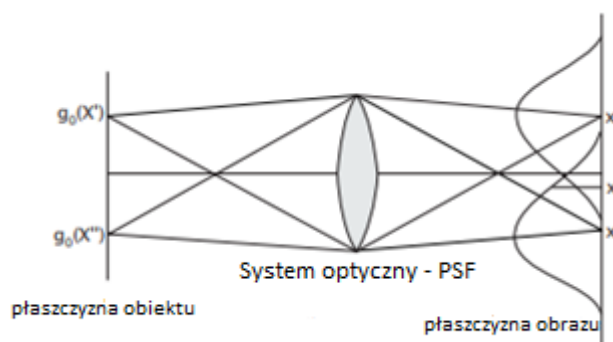
Obecnie można pozyskiwać obrazy z szerokiego zakresu narzędzi. Każdy z tych odbiorników jest w stanie przetwarzać pewien rodzaj energii na formę wizualną.

4.1 Jak powstaje obraz?

Proces tworzenia obrazu jest określany jako suma pewnych kluczowych elementów. Ogólnie obraz można przedstawić jako matematyczny model składający się z funkcji obiektu, procesu przechwytywania (PSF point spread function) oraz z dodatkowego szumu. Na schemacie przedstawionym na rys. 3 za PSF odpowiada system optyczny. PSF jest określana przez fizyczne własności sprzętu użytego do obrazowania np. wielkość i kształt soczewek.

$$\text{Obraz} = \text{PSF} * \text{funkcja obiektu} + \text{szum} \quad (4.1)$$

gdzie: PSF opisuje, jak informacje o funkcji obiektu są rozpowszechniane ze względu na otrzymane dane, funkcja obiektu przedstawia obiekt lub scenerie oraz jak jest odbijane światło od tej struktury do urządzenia obrazującego, szum jest to funkcja, która opisuje wszystkie zakłócenia, które wystąpiły podczas rejestracji obrazu, * jest to operator konwolucji, która działa na zasadzie splotu dwóch funkcji.



Rys 3. Formowanie obraz przez konwolucję z wykorzystaniem funkcji rozmycia punktów [2]

4.2 Filtracja cyfrowa

Filtry cyfrowe stanowią podstawowe narzędzie przy przetwarzaniu obrazów. Operacje przeprowadzane przez filtry są kontekstowe. Oznacza to, że aby uzyskać wartość dla jednego piksela, należy wykonać obliczenia dla punktów z nim sąsiadujących. Polega to na wyznaczeniu wartości funkcji dla piksela na obrazie oryginalnym oraz wartości w jego otoczeniu. Zwykle otoczenie (nazywane też maską,

rys. 4) przyjmuje formę kwadratu, aby można było łatwo wskazać punkt centralny. Ze względu na kontekstowość filtracja nie może przebiegać na krawędziach obrazu, gdyż zabraknie wartości dla argumentów przy ścianie maski. Mimo powyższego ograniczenia operacje filtracji są bardzo powszechnie wykorzystywane przy przetwarzaniu obrazów. Filtry cyfrowe mają istotny wpływ na informacje na obrazie i pozwalają pozbyć się zakłóceń, takich jak szum czy rozmycie. Pod względem matematycznym filtr jest funkcją przekształcającą obraz piksel za pikselem. Właściwości filtru określane są ze względu na typ filtru. Wyróżnia się filtry liniowe oraz nieliniowe.

$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$
$\frac{1}{3}$	$\frac{1}{3}$	$\frac{1}{3}$

Rys. 4. Przykładowa kwadratowa maska 3x3 [5]

4.2.1 Filtry liniowe

W filtrach liniowych nowa wartość piksela jest definiowana jako liniowa kombinacja wartości pikseli w jego sąsiedztwie. Liniowa kombinacja sąsiednich pikseli jest określana przez jądro filtracyjne nazywane maską. Podczas filtracji sukcesywnie pozycjonuje się maskę tak, by położenie jej środka pokrywało się z lokalizacją każdego piksela docelowego. Mnoży się piksele pod jądrem przez odpowiadające im wagi na masce, a następnie sumuje się otrzymane wartości i kopiuje do tych samych lokalizacji na nowym obrazie. Można powiedzieć, że filtrowanie liniowe działa na tej samej zasadzie co konwolucja, oba te terminy można stosować zamiennie.

W przypadku pikseli znajdujących się na granicy obrazu, czyli gdy wartości maski wychodzą poza krawędzie obrazu, można działać na kilka sposobów:

- pozostawić bez zmian piksele znajdujące się na granicy,
- filtrację przeprowadzać tylko na pikselach, które leżą w obrębie granicy i regulować działanie filtra,

- wypełnić brakujące piksele poprzez odzwierciedlenie wartości nad granicą i dalsze filtrowanie.

Filtry wykorzystujące funkcje liniowe muszą spełniać dwa warunki:

- addytywność

$$\varphi(f + g) = \varphi(f) + \varphi(g) \quad (4.2)$$

- jednorodność

$$\varphi(\lambda f) = \lambda \varphi(f) \quad \lambda \in R \quad (4.3)$$

Filtry liniowe są powszechnie stosowane ze względu na łatwość w implementacji i intuicyjność przekształceń.

4.2.2 Filtry nieliniowe

Filtry nieliniowe działają na podobnej zasadzie jak filtry liniowe. Maską również jest przesuwana po obrazie jak w przypadku filtrów liniowych. Różnica polega na tym, że przefiltrowana wartość, będzie otrzymywana w wyniku nieliniowych operacji na sąsiednich pikselach. Filtry nieliniowe są trudne w realizacji i w związku z tym mniej popularne od filtrów liniowych. Jednakże filtry nieliniowe pozwalają dokładniej lokalizować krawędzie niż filtry liniowe. Dzięki nieliniowym przekształceniom odtworzony obraz ma dużo lepiej podkreślone kontury.

4.3 Konwolucja

Konwolucja nazywana również splotem, opisuje podstawowe przekształcenia w rekonstrukcji obrazów. Dzięki konwolucji jest możliwa filtracja liniowa. Konwolucję definiuje się następująco:

$$g(x) = (f \times h)(x) = \int_{-\infty}^{+\infty} f(x - t)h(t) dt \quad (4.4)$$

gdzie: f, h – splatane funkcje

Działanie konwolucji polega na zasadzie nakładającej się całki. Odwrócona wersja jednej z funkcji podstawowych jest systematycznym przemieszczeniem, a całka z jej wyniku (obszaru nakładania) jest obliczana na każdej zmianie pozycji. Wszystkie przesunięcia, w których obszar nakładania nie jest zerowy, określają zasięg konwolucji.

4.3.1 Konwolucja dyskretna

W cyfrowej analizie obrazów sygnały są dyskretne a nie ciągłe. Konwolucję dla dwuwymiarowego dyskretnego obrazu przedstawia wzór:

$$I'(m,n) = (w \times I)(m,n) = \sum_{i,j \in M} I(m-i, n-j)w(i,j) \quad (4.5)$$

gdzie $w(i,j)$ to wartości współczynników maski, $I'(m,n)$ to obraz po konwolucji, (m,n) to współrzędne pikseli

Filtry definiuje się poprzez tablice wartości $w(i,j)$. Współczynniki $w(i,j)$ razem z pikselami obrazu $I(m-i, n-j)$, znajdującymi się w obrębie maski M umieszczonymi wokół współrzędnych punktu (m,n) pozwalają obliczyć wartości $I'(m,n)$ na obrazie wynikowym. Ważne jest, aby współczynniki $w(i,j)$ były liczbami całkowitymi, co umożliwia kilkunastokrotne zmniejszenie ilości obliczeń w stosunku do ilości obliczeń na liczbach zmiennoprzecinkowych. Dla prostych zakłóceń wartości współczynników są większe lub równe zero, można wtedy zastosować następujący wzór:

$$I'(m,n) = \frac{1}{\sum_{(i,j) \in M} w(i,j)} \sum_{i,j \in M} I(m-i, n-j)w(i,j) \quad (4.6)$$

gdzie oznaczenia są analogiczne, jak dla poprzedniego wzoru

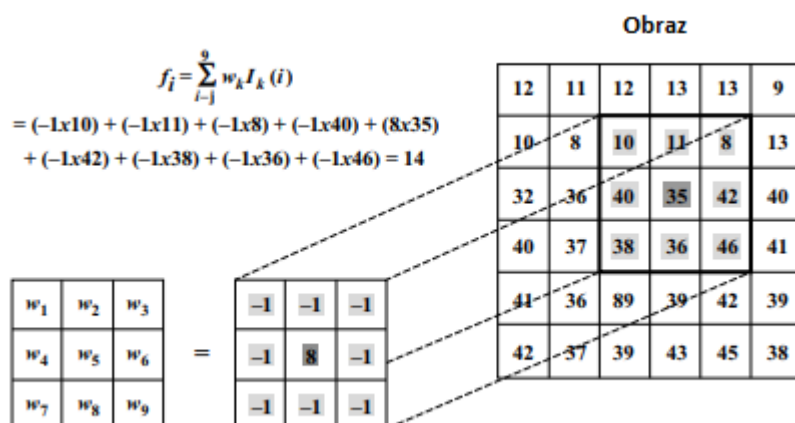
W przypadku szumów, dla których współczynniki mogą przyjmować wartości dodatnie i ujemne należy odwołać się do wartości $\min I'(m,n)$ i $\max I'(m,n)$. Przedstawia to poniższy wzór:

$$I''(m,n) = \frac{I'(m,n) - \min I'(m,n)}{\max I'(m,n) - \min I'(m,n)} \quad (4.7)$$

Wzór (4.7) można stosować dla maski ze współczynnikami dodatnimi i ujemnymi. Jednakże pod względem praktycznym jest on niewygodny, dlatego wprowadzono zapis uproszczony wykorzystujący kolejne numery współczynników:

$$\begin{aligned} I'(m,n) = & w_1 I(m-1, n-1) + w_2 I(m-1, n) + w_3 I(m-1, n+1) \\ & + w_4 I(m, n-1) + w_5 I(m, n) + w_6 I(m, n+1) \\ & + w_7 I(m+1, n-1) + w_8 I(m+1, n) + w_9 I(m+1, n+1) \end{aligned} \quad (4.8)$$

Zasadę działania dyskretnej konwolucji (4.8) przedstawiono na rysunku 5.



Rys. 5. Dyskretna konwolucja. Punkt 35 po zastosowaniu maski przyjmuje wartość 14 [3]

4.4 Kwantyzacja

Obraz w formie cyfrowej może mieć obszary o pewnym jednolitym odcieniu szarości. Piksele na tych obszarach mają taką samą wartość w skali szarości. Kwantyzacja polega na odwzorowaniu tych pikseli na pewną ograniczoną liczbę Q dyskretnych poziomów szarości. Numer wymaganego poziomu kwantyzacji jest określany przez dwa kryteria.

Pierwsze kryterium mówi, że żadne szare wartości nie powinny być rozpoznawalne, tak jak nie można rozróżnić pojedynczych pikseli na obrazach cyfrowych. Ogólnie dane obrazu są skwantowane do 256 poziomów szarości, gdzie każdy piksel zajmuje 8 bitów.

Drugie kryterium jest powiązane z jakością obrazu. Określa ilość bitów wymaganych do opisu poziomów szarości w zależności od przeznaczenia obrazu. Większa ilość bitów oznacza więcej informacji na obrazie. Na przykład w wizji maszynowej są wykorzystywane tylko dwa poziomy kwantyzacji (obraz binarny), ale z kolei w diagnostyce medycznej standardowa 8-bitowa rozdzielczość byłaby niewystarczająca. Stosuje się wtedy co najmniej 24-bitową rozdzielczość.

4.5 Szum w obrazach

Szum w obrazach jest przypadkową zmiennością jasności, albo informacji o kolorze w obrazie otrzymanym ze skanera, albo aparatu cyfrowego. Szum może też wynikać z ziarnistości filmu, albo zakłóceń na detektorze fotonów. Można powiedzieć, że szum jest niepożądanym efektem ubocznym przy przechwytywaniu obrazu. Typy szumów są następujące:

- szum gaussowski,
- szum „pieprz i sól”,
- szum śrutowy (szum Poissona),
- szum plamkujący,

Szum gaussowski jest standardowym modelem szumu wzmacniającego. Jest on niezależny na każdym pikselu oraz niezależny od intensywności sygnału. W kolorowych kamerach większe wzmocnienie przypada na kolor niebieski, niż na zielony czy czerwony.

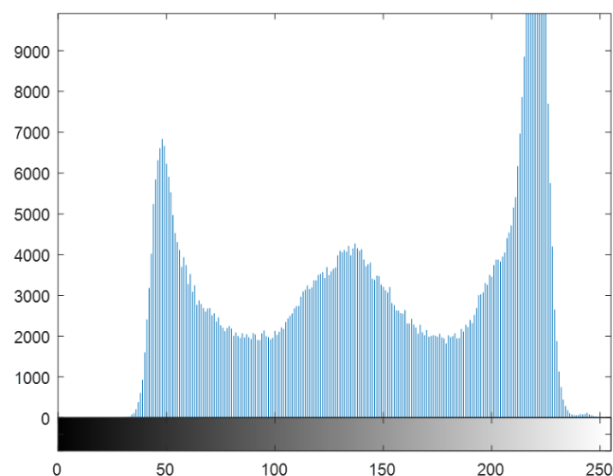
Obraz który zawiera szum „pieprz i sól” będzie miał czarne piksele w jasnych regionach i jasne piksele w ciemnych regionach. Taki typ szumu jest spowodowany przez martwe piksele analogowo-cyfrowe. Martwe piksele są to takie piksele, które nie przyjmują oczekiwanej wartości (barwy), albo przyjmują wartości losowe.

Szum Poissona, nazywany szumem śrutowym, jest typem szumu elektronicznego, który pojawia się, gdy liczba cząstek przenoszących energię (elektrony lub fotony) na urządzenie optyczne jest tak mała, że powoduje wykrywalne wahania statystyczne.

Szum plamkujący jest to szum granulacyjny, istniejący z natury i pogarszający jakość obrazu na działającym radarze. Wynika z przypadkowych wahań sygnału zwrotnego z obiektu, który nie jest większy niż pojedynczy element przetwarzania obrazu. Powoduje to zwiększenie średniego poziomu szarości na obrazie lokalnym.

5. Histogramy

Histogram informuje ile pikseli na obrazie przypada na każdą wartość w skali szarości ($[0,255]$). Po znormalizowaniu takiego wykresu, tak aby suma wszystkich częstotliwości w dopuszczalnym zakresie była jednolita, otrzymuje się funkcję gęstości prawdopodobieństwa wystąpienia danej wartości piksela na obrazie. Obserwując histogram można m.in. rozpoznać kontrast na obrazie czy wychwycić podstawowe różnice w kolorze komponentów. Dla zwykłego obrazu w skali szarości histogram konstruuje się poprzez zliczanie ilości występowania każdej wartości w skali szarości. Przykładowy histogram na rys. 6 ma 3 szczyty. Pierwszy szczyt oznacza ciemną karoserię samochodu, drugi szczyt znajduje się w środku histogramu i odzwierciedla ulicę oraz ludzi za samochodem, trzeci szczyt symbolizuje jasne tło.



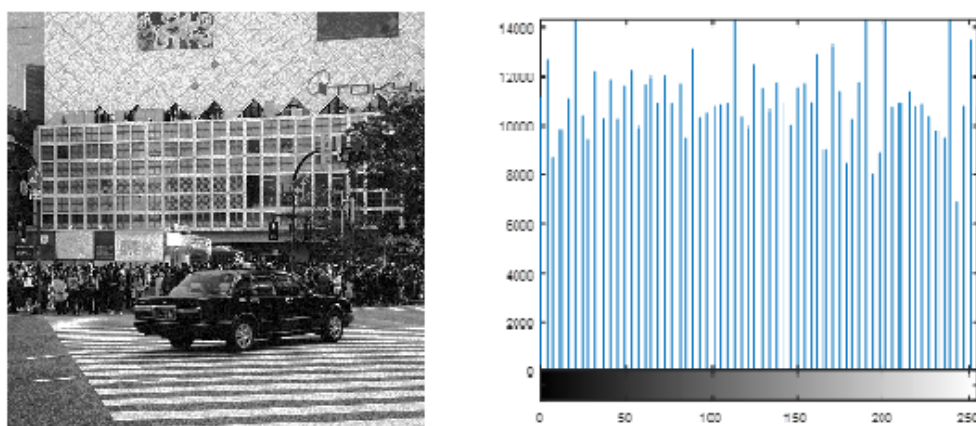
Rys. 6. Przykładowy obraz i jego histogram [8]

5.1 Wyrównywanie histogramów

Jedną z operacji manipulowania histogramami, wykorzystywaną przy rekonstrukcji obrazów, jest wyrównywanie histogramów. Pod względem teoretycznym wygląda to tak, że zakłada się obraz wejściowy w skali szarości i oznacza jako $I(x)$. Jeśli zmienna x jest ciągła i znormalizowana w przedziale $[0,1]$, można rozważyć znormalizowany histogram, jako funkcję gęstości prawdopodobieństwa (PDF) $p_x(x)$. Funkcja ta pozwala określić prawdopodobieństwo wystąpienia danych wartości w pobliżu x . Podobnie można otrzymać obraz wyjściowy po wyrównaniu histogramu.

5.1.1 Wyrównywanie histogramów w praktyce

W praktyce wyrównywanie histogramów to w pełni zautomatyzowana i obliczeniowo prosta procedura. Transformacja intensywności $y=f(x)$ zależy tylko od histogramu wejściowego. Konsekwencją modelowania histogramu jest modyfikacja jego zakresu dynamicznego. Dzięki temu liczba pikseli przypadających na poszczególne poziomy w skali szarości przed wyrównaniem, przyjmuje po wyrównaniu zbliżoną do siebie wartość. Przy wyrównaniu jest zastosowane monotoniczne, nieliniowe odwzorowanie dla zachowania jednolitego kształtu histogramu na obrazie wyjściowym. Na histogramie na rys. 7 można zauważyć efekt wyrównania. Histogram został rozrzedzony, a niektóre barwy zostały wzmocnione.



Rys 7. Obraz i histogram po wyrównaniu [8]

5.2 Histogramy obrazów kolorowych

Poprawianie kontrastu na obrazach kolorowych jest nieco bardziej złożone, niż na obrazach w skali szarości. Mogłoby się wydawać, że wyrównanie histogramu będzie się odbywać na każdym z trzech kanałów (R,G,B) oryginalnego obrazu. Jednakże wartości RGB zawierają zarówno informacje o intensywności, jak i chromatyczności obrazu. Poprawa kontrastu obrazu z wykorzystaniem histogramu działa na takiej zasadzie, że najpierw obraz jest przekształcany z RGB na HSV (barwa, nasycenie, wariancja), następnie stosuje się wyrównanie w odniesieniu do składnika nasycenia i na końcu przekształca wynik z powrotem na RGB. Efekt przedstawiono na rys. 8.



Rys 8. Poprawa kontrastu obrazu kolorowego przez wyrównanie histogramu w Matlabie [9]

6. Filtrowanie szumów

Usuwanie szumów jest jednym z głównych zastosowań filtrów liniowych i nieliniowych. W dalszej części tego rozdziału scharakteryzowano kilka podstawowych filtrów służących do rekonstrukcji obrazów.

6.1 Filtr uśredniający

Filtrowanie linowe pozwala usunąć niektóre rodzaje szumu, a jednym z podstawowych filtrów liniowych jest filtr uśredniający. Filtr uśredniający pozwala usunąć ziarnistość z obrazu, gdyż wartość piksela docelowego jest wyznaczana na podstawie średniej wartości pikseli w jego sąsiedztwie (włączając przy tym piksel w jądrze). Dzięki temu można wyeliminować piksele, które są niereprezentowane dla otoczenia. Stosowanie tej filtracji powoduje, że lokalne wahania spowodowane przez ziarna są zmniejszone. Wadą filtru uśredniającego jest uszkodzenie krawędzi obrazu. Przeważnie stosowane są maski 3x3, większe maski są stosowane do bardziej rygorystycznego wygładzenia obrazu.

6.2 Filtr medianowy

Filtr medianowy należy do nieliniowych technik filtracji stosowanych do usuwania szumu. Przetwarzanie obrazu filtrem medianowym jest wstępnym etapem dla późniejszych przekształceń, takich jak np. wykrywanie krawędzi. Filtracja medianowa jest dość powszechnie stosowana dlatego, że w pewnych warunkach zachowuje krawędzie, usuwając przy tym szum. Działanie tego filtru jest zbliżone do działania filtru uśredniającego, tylko że w tym przypadku liczy się medianę z wartości znajdujących się pod maską poprzez posortowanie wartości w porządku numerycznym. Łatwo zauważyć, że w przypadku, gdy maska ma nieparzystą liczbę wyników jest po prostu środkowa wartość, dla parzystej liczby wartości uzyskuje się dwie mediany.

Przewagą filtru medianowego nad liniowymi jest to, że może usunąć bardzo dużą wartość szumu wejściowego. Mediana jest też bardziej solidna od zwykłej średniej, więc niepożądany piksel w sąsiedztwie nie wpłynie znacząco na wynik. Dodatkowo filtr medianowy nie tworzy nowych pikseli podczas filtracji.

6.3 Filtr Gaussowski

Filtr gaussowski jest bardzo ważnym filtrem, za pomocą którego można m.in. definiować rozmycie, odtwarzać obraz czy też nakładać szum. Do filtracji jest stosowana dyskretna maska zachowująca promieniowo symetryczny kształt. Przykład takiej maski jest przedstawiony na rys. 9. Największą wartość ma element w centrum maski, waga pozostałych zmniejsza się wraz z odległością od punktu centralnego.

1	2	1
2	4	2
1	2	1

Rys 9. Przykładowy filtr Gaussa [10]

Maska pochodzi z ciągłej dwuwymiarowej funkcji gaussowskiej.

$$f(x, y) = \frac{1}{2\pi\sigma^2} \exp\left(-\frac{x^2+y^2}{2\sigma^2}\right) \quad (6.1)$$

Dyskretna aproksymacja funkcji (6.1) jest określana przez dwa parametry. Pierwszy z nich określa rozmiar maski, a drugi - definiuje wartość σ , czyli odchylenia standardowego. Stosowanie filtru gaussowskiego ma również efekt wygładzający, tak jak w przypadku filtru uśredniającego.

7. Rekonstrukcja obrazów

Rekonstrukcja obrazów opiera się na próbie polepszenia jakości obrazu na podstawie wiedzy o procesach fizycznych, które umożliwiają uzyskanie niezbędnych informacji wymaganych do rekonstrukcji. Formowanie obrazów można traktować jako proces transformacji danych wejściowych na dane wyjściowe. Dane wejściowe traktuje się jako idealny obraz. Obraz idealny zostaje następnie poddany zaszumieniu i rozmyciu. Rekonstrukcja będzie polegać na jak najlepszym odtworzeniu obrazu uszkodzonego w stosunku do stanu pierwotnego, poprzez np. zniwelowanie rozmycia lub redukcję szumu. Liniowe dwuwymiarowe systemy obrazowe będą reprezentowane przez wspomnianą w poprzednich rozdziałach konwolucję.

$$g(x, y) = \iint f(x', y') h(x - x', y - y') dx' dy' + n(x, y) \quad (7.1)$$

$$g(x, y) = f(x, y) ** h(x, y) + n(x, y) \quad (7.2)$$

gdzie $**$ używa się do notacji dwuwymiarowej funkcji, $h(x, y)$ jest funkcją rozmycia punktów (PSF) lub odpowiedzią impulsową, a $n(x, y)$ jest addytywnym szumem. Rekonstrukcja polega na oszacowaniu rozkładu wejściowego $f(x', y')$ używając zmierzonego wyjścia $g(x, y)$ i wiedzy o posiadanym PSF $h(x - x', y - y')$ oraz szumie $n(x, y)$. Proces rekonstrukcji obrazu wejściowego $f(x', y')$ nazywa się dekonwolucją. Powyższe równanie nie jest oczywiście jedynym sposobem przetwarzania obrazów, niektóre sytuacje wymagają bardziej zaawansowanych środków, takich jak warianty przestrzenne czy modele nieliniowe.

PSF ma zazwyczaj stałą i deterministyczną wartość i wynika z błędów sprzętu, użytego do procesu obrazowania. Na przykład w sprzęcie optycznym, PSF jest określona przez fizyczne własności i kształt soczewek, w systemach medycznych wykrywających promieniowanie poprzez połączenie kolimatora ze scyntylatorem, który umożliwia wykrywanie promieniowania z pewną dokładnością. Można więc powiedzieć, że PSF jest czymś, nad czym można mieć kontrolę i co staje się stałe dla zaprojektowanego systemu optycznego. Właśnie dlatego, że PSF jest wynikiem procesu inżynierskiego można posiadać o niej pewną wiedzę, która pomaga w procesie rekonstrukcji. Jak już wspomniano w punkcie 4.5, szum powoduje przypadkowe i niepożądane wahania zarejestrowanego sygnału. Najważniejszym elementem przy opisie szumu jest to, że nad szumem nie ma kontroli i nie można go przewidzieć. Szum pochodzi z fizycznego procesu detekcji i ma wiele specyficznych form. Bez względu na fizyczny proces

powstawania obrazu, cechą wspólną jest nieprzewidywalny charakter wahań. Choć nie można znać konkretnej wartości i rodzaju szumu, można je zamodelować i wykorzystać w metodach rekonstrukcji.

8. Zastosowanie algorytmu Lucy-Richardsona do rekonstrukcji obrazu

Algorytm dekonwolucji Lucy-Richardsona jest popularny w dziedzinie astronomii oraz medycyny. Algorytm wywodzi się z twierdzenia Bayesa i został odkryty we wczesnych latach siedemdziesiątych przez Richardsona i Lucy [2]. Działanie algorytmu LR najlepiej zrozumieć dzięki przytoczeniu prostego algorytmu iteracyjnego, a potem poszerzyć go na metodę Lucy-Richardsona. Wykorzystując liniowe równanie obrazowania (7.2), można stwierdzić, że szum $n(x, y)$ jest różnicą między wartością wyjściową, a nieznaną wartością wejściową spletaną z PSF $h(x, y)$

$$n(x, y) = g(x, y) - f(x, y) ** h(x, y) \quad (7.3)$$

Aby dobrze oszacować wartość wejściową szum $n(x, y)$ musi mieć pomijalną wartość. Dlatego najlepsze odwzorowanie powinno wyglądać następująco:

$$g(x, y) - f(x, y) ** h(x, y) = 0 \quad (7.4)$$

Jeśli doda się rozkład wejściowy $f(x, y)$ do obu stron równania to otrzyma się

$$f(x, y) = f(x, y) + [g(x, y) - f(x, y) ** h(x, y)] \quad (7.5)$$

To równanie można potraktować jako iteracyjną procedurę, gdzie wartość wejściowa (lewa strona równania) jest otrzymywana poprzez sumę poprzedniego oszacowania (prawa strona) i korekcji (w nawiasach). Równanie można zatem zapisać w formie procedury iteracyjnej w następujący sposób.

$$f_{i+1}(x, y) = f_i(x, y) + [g(x, y) - f_i(x, y) ** h(x, y)] \quad (7.6)$$

Jeśli obraz nie jest mocno zdegradowany, wartość wyjściowa nie będzie się różnić zbytnio od wartości wejściowej. Równanie nigdy nie zostanie spełnione dokładnie, chyba że wartość szumu wynosi zero.

Cechy metody LR są następujące:

- zakłada się, że PSF jest znane,
- iteracja jest zbieżna z maksymalnym prawdopodobieństwem dla statystyk Poissona,
- metoda LR odtwarza obraz jako nieujemny,
- odtworzone obrazy są odporne na małe błędy w PSF,
- rekonstrukcje LR wymagają odpowiedniego czasu,

Można powiedzieć, że i -ty piksel wejściowy w każdej iteracji ma wartość f_i co jest powiązane z i -tym pikselem dla wartości wyjściowej g_i

$$g_i = \sum_j h_{ij} f_j \quad (7.7)$$

gdzie sumowanie po j oznacza wkład każdego piksela wejściowego jako konwolucje z h_{ij} do otrzymanego piksela wyjściowego. Iteracyjna forma równania LR przyjmuje postać:

$$f_j^{t+1} = f_j^t \left(\frac{q}{f_j^{t**h}} ** \hat{h} \right) \quad (7.8)$$

gdzie $**$ to symbol konwolucji.

8.1 Implementacja funkcji dekonwolucji algorytmem Lucy-Richardsona

Działanie funkcji dekonwolucji algorytmem Lucy-Richardsona w dziedzinie przestrzennej zostało przedstawione na rys. 10. Funkcja wykonuje zaimplementowane równanie (7.8) w pętli dla kolejnych iteracji.

Algorithm 1 Funkcja dekonwolucji algorytmem Lucy-Richardsona w dziedzinie przestrzennej

```

1: procedure DEKONWOLUCJARL(obraz, PSV, iteracje)
2:   %Ze względu na poprawność numeryczną:
3:   obraz ← double(obraz)
4:   PSV ← double(obraz)
5:   obrazDyskretny ← obraz
6:   %przestrzennie odwrócona PSF dla obrazu dyskretnego:
7:   odwrotnyPSV ← odwroc(PSV)
8:   for  $i$  w zakresie iteracje do
9:     konwolucja ← konwolucja obrazuDyskretnego i PSV
10:    rozmycie ← stosunek obrazu do konwolucji
11:    bład ← konwolucja rozmycia i odwrotnegoPSV
12:    obrazDyskretny ← iloczyn obrazuDyskretnego i błędu
13:  end for
14:  %Zwrocenie obrazu dyskretnego
15:  wynik ← obrazDyskretny
16: end procedure

```

Rys. 10. Pseudokod dla funkcji w dziedzinie przestrzennej [11]

Funkcja dekonwolucji w dziedzinie przestrzennej została zaimplementowana w Matlabie (rys. 11).

```
function result = dekonwolucjaRL(obraz, FRP, iter)
    % obrazy muszą być w formacie double by wykonać konwolucje
    obraz = double(obraz);
    FRP = double(FRP);
    obrazdys = obraz; % wprowadzenie na obraz dyskretny
    OdwrFRP = FRP(end:-1:1,end:-1:1); % przestrzennie odwrócona PSF
    %dla obrazu dyskretnego

    for i= 1:iter
        konwolucja = convn(obrazdys,FRP,'same');
        Rozmycie2 = obraz./konwolucja;
        blad = convn(Rozmycie2,OdwrFRP,'same');
        obrazdys = obrazdys.* blad;
    end

    result = obrazdys;
end
```

Rys. 11. Fragment kodu źródłowego dla funkcji w dziedzinie przestrzennej

Na rys. 12 przedstawiono funkcje realizującą algorytm Lucy-Richardsona w dziedzinie częstotliwości. Działanie funkcji polega na przekształceniu obrazu i funkcji rozmycia (PSF) do dziedziny częstotliwości. Następnie funkcja wykorzystuje numeryczną implementację wzoru (7.8) w pętli iteracyjnej.

Algorithm 1 Funkcja dekonwolucji algorytmem Lucy-Richardsona w dziedzinie częstotliwościowej

```
1: procedure CZESTOTLIWOSCRL(obraz)
2:   obrazWPetli  $\leftarrow$  obraz
3:   %zastosowanie funkcji transferu optycznego:
4:   OFP  $\leftarrow$  zastosowanie funkcji transferu optycznego
5:   for i w zakresie iteracje do
6:     uklad  $\leftarrow$  FFTobrazuWPetli
7:     splot  $\leftarrow$  splot układu i OFP
8:     transformacjaodw  $\leftarrow$  odwrotna FFT dla splotu
9:     stosunek  $\leftarrow$  iloraz obrazu i transformacjiodw(per element)
10:    ukladStosunku  $\leftarrow$  transformacja Fouriera dla stosunku
11:    wynik  $\leftarrow$  iloczyn OFP i układuStosunku(per element)
12:    transformataWyniku  $\leftarrow$  odwrotny FFT dla wyniku
13:    obrazWPetli  $\leftarrow$  iloczyn transformatyWyniku i obrazuWPetli
14:  end for
15:  %Zwrocenie znormalizowanego obrazu
16:  wynik  $\leftarrow$  wartosc bezwzględna z wyniku
17: end procedure
```

Rys. 12. Pseudokod dla funkcji w dziedzinie częstotliwości [11]

Realizacja funkcji w dziedzinie częstotliwości w Matlabie (rys. 13)

```
function result = czestotRL(obraz, FRP, iter)
    obraziter = obraz; %wczytanie obrazu do iteracji
    OFP = psf2otf(FRP,size(obraz)); %OFP optyczna funkcja przenoszenia
    %z ang. OTF-optical transfer function, zmiana FRP na OTF

    for i=1:iter
        uklad = fft2(obraziter); %przekształcenie do dziedziny częstotliwości
        splot = OFP.*uklad;
        transformacjaodw = ifft2(splot); %odwrotna transformacja
        %fouriera(czestotliwosciowa)
        stosunek = obraz./transformacjaodw;
        ukladstosunku = fft2(stosunek);
        wynik = OFP .* ukladstosunku;
        transfwyniku = ifft2(wynik);
        obraziter = transfwyniku.*obraziter;
    end
    result = abs(obraziter);
end
```

Rys. 13. Fragment kodu źródłowego dla funkcji w dziedzinie częstotliwości

8.2 Rekonstrukcja obrazu algorytmem Lucy-Richardsona

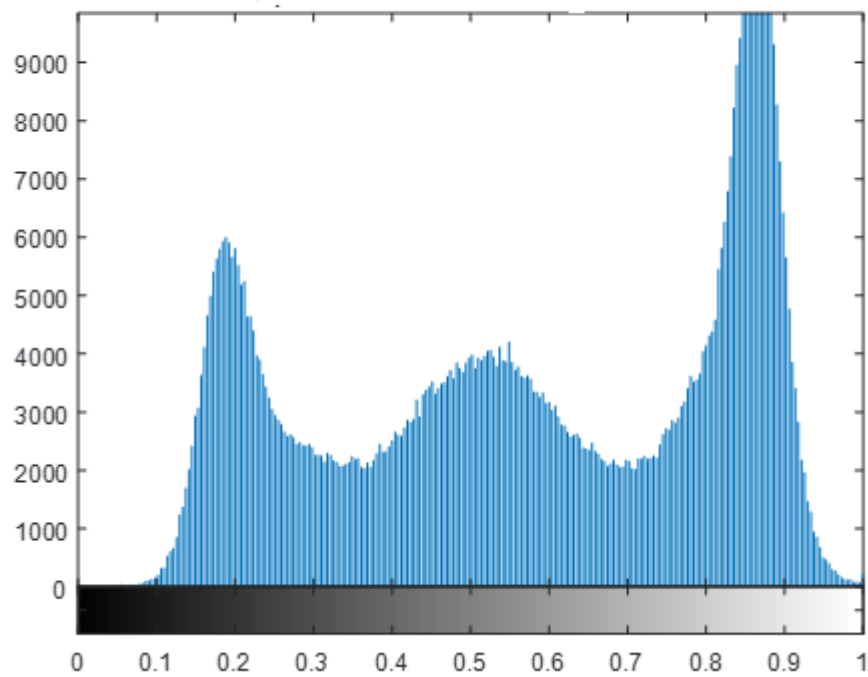
Obraz o rozmiarze 512x512 pikseli (rys. 14) rozmyto filtrem gaussowskim o rozmiarze 5x5 i odchyleniu standardowym 5 (rys.15). Rozmyty obraz zaszumiono szumem gaussowskim o średniej równej 0 i amplitudzie szumu $v = 0,002$. Otrzymano wtedy obraz jak na rys. 16. Zaimplementowane zostały dwie funkcje wykonujące rekonstrukcje w dwóch dziedzinach. Pierwsza rekonstrukcja została przeprowadzona w dziedzinie częstotliwości (Fouriera) dla 15 iteracji (rys. 17). Drugą rekonstrukcję przeprowadzono w dziedzinie przestrzennej (rys. 18), również dla 15 iteracji. Następnie spośród dwóch wyżej wymienionych rekonstrukcji wybrano tę, która daje lepsze rezultaty. Lepszej rekonstrukcja została wybrana na podstawie jakości odtworzonego obrazu oraz czasu wykonywania operacji. Lepszy rezultat otrzymano dla dziedziny przestrzennej. Dla odtworzonego obrazu w dziedzinie przestrzennej przeprowadzono kontrolę wzmocnienia szumu poprzez tłumienie (rys. 19). Tłumienie wykonano z użyciem gotowych funkcji w Matlabie ze względu na trudności w implementacji. Wartość parametru tłumienia wynosi 0,178885. Kolejnym zastosowanym parametrem była waga. Zadaniem wagi było zredukowanie powielających się krawędzi (rys. 20). Waga została wykonana z użyciem gotowych funkcji z własnymi parametrami

a) Obraz oryginalny



Photo by Jon Tang on Unsplash, link - <https://unsplash.com/photos/UC0YQ8DmU>

b) Histogram obrazu oryginalnego

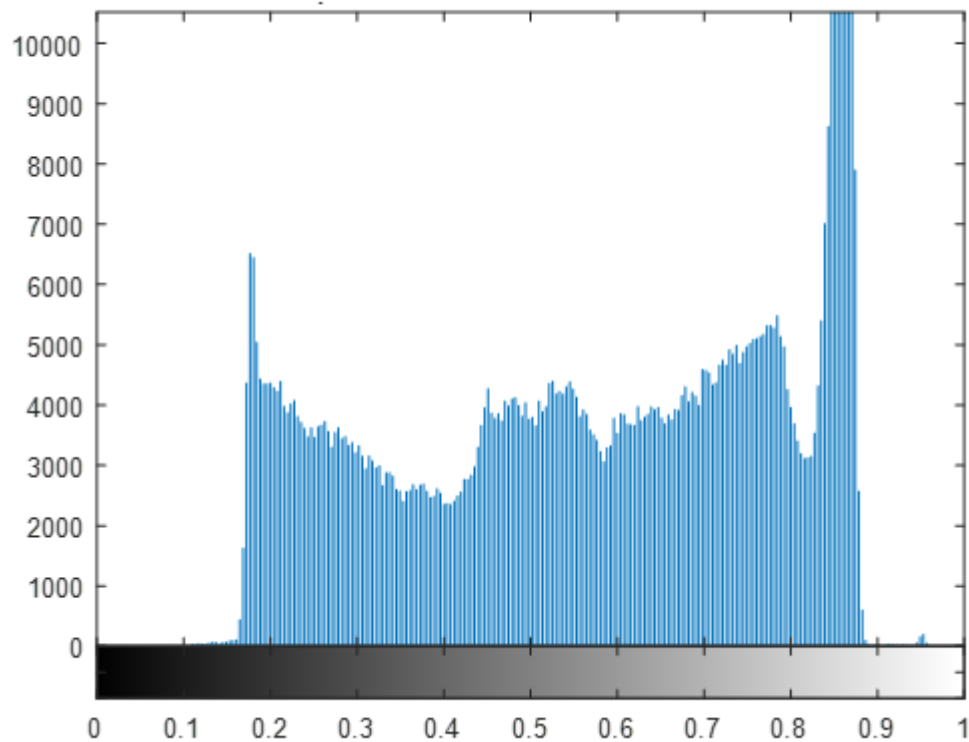


Rys 14. Obraz oryginalny i jego histogram

a) Obraz rozmyty



b) Histogram rozmytego obrazu

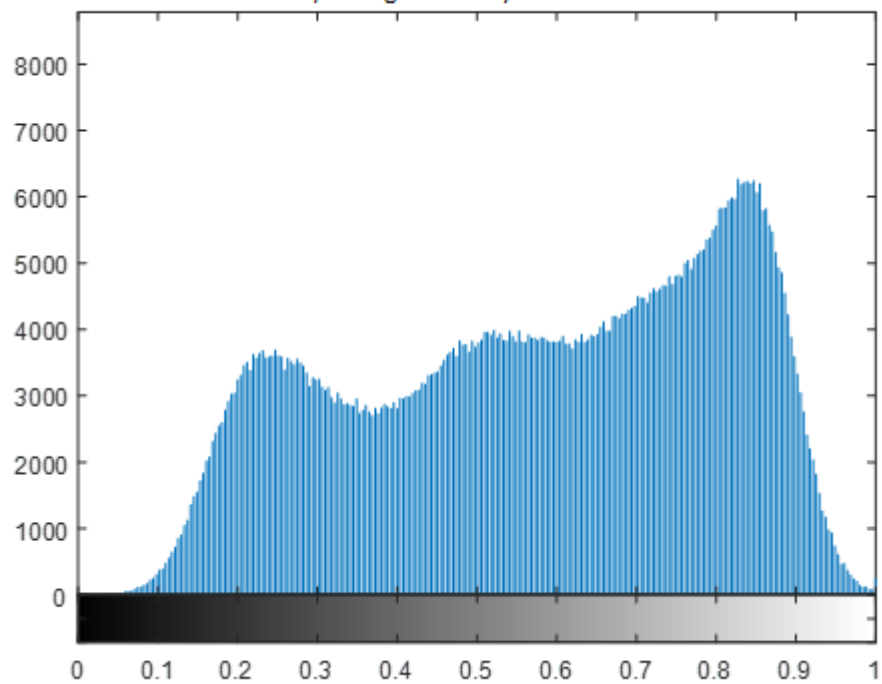


Rys 15. Obraz po nałożonym rozmyciu ze swoim histogramem

a) Rozmycie z szumem na obrazie początkowym



b) Histogram rozmycia z szumem

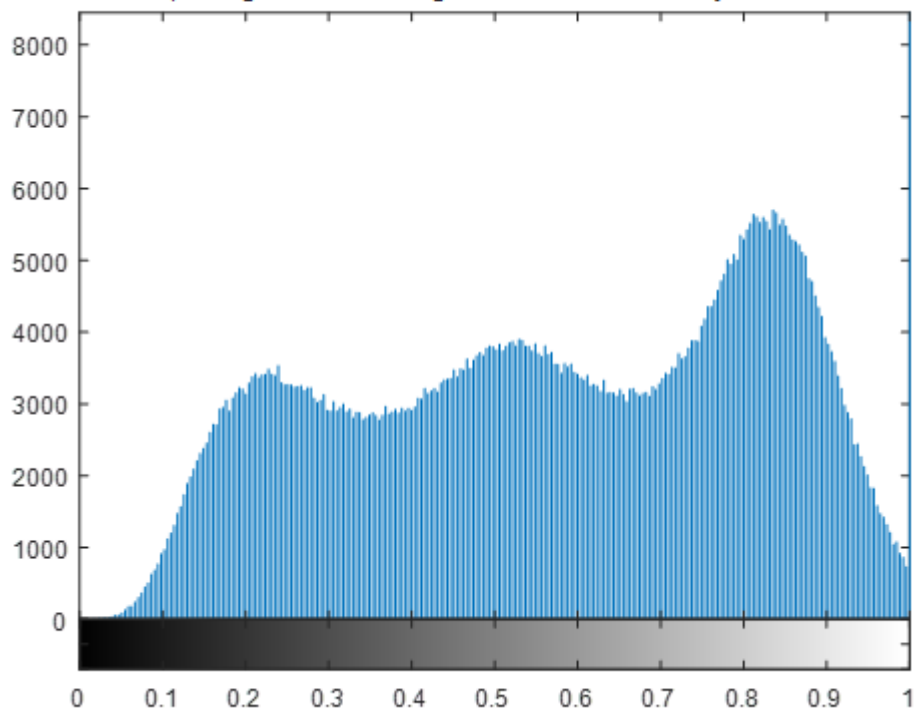


Rys 16. Obraz rozmyty i zaszumiony z histogramem

a) Odtworzony obraz w dziedzinie częstotliwości



b) Histogram odtworzonego obrazu w dziedzinie częstotliwości

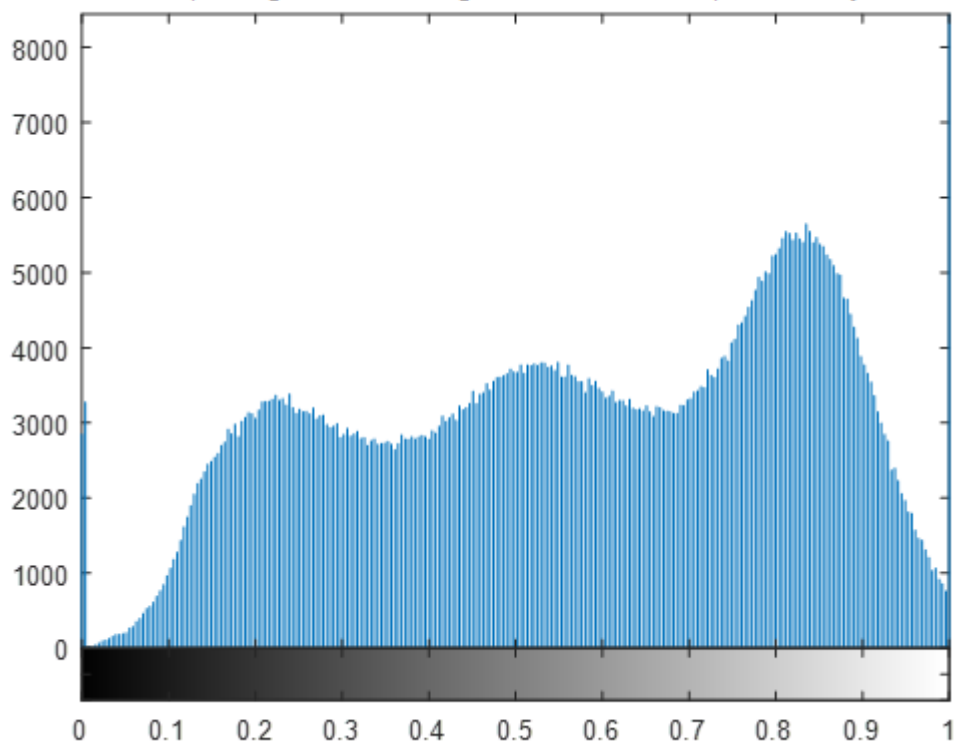


Rys 17. Rekonstrukcja obrazu w dziedzinie częstotliwości po 15 iteracjach. Czas przetwarzania 1,839857 sekund

a) Odtworzony obraz w dziedzinie przestrzennej



b) Histogram odtworzonego obrazu w dziedzinie przestrzennej

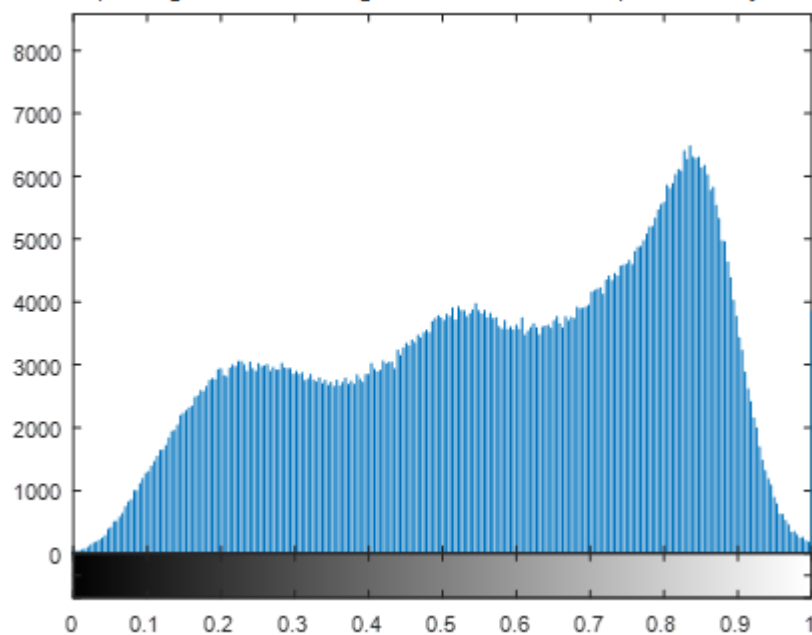


Rys 18. Rekonstrukcja w dziedzinie przestrzennej obrazu po 15 iteracjach. Czas przetwarzania 0,572242 sekund

a) Odtworzony obraz z tłumieniem, po 15 iteracjach



b) Histogram odtworzonego obrazu z tłumieniem po 15 iteracjach

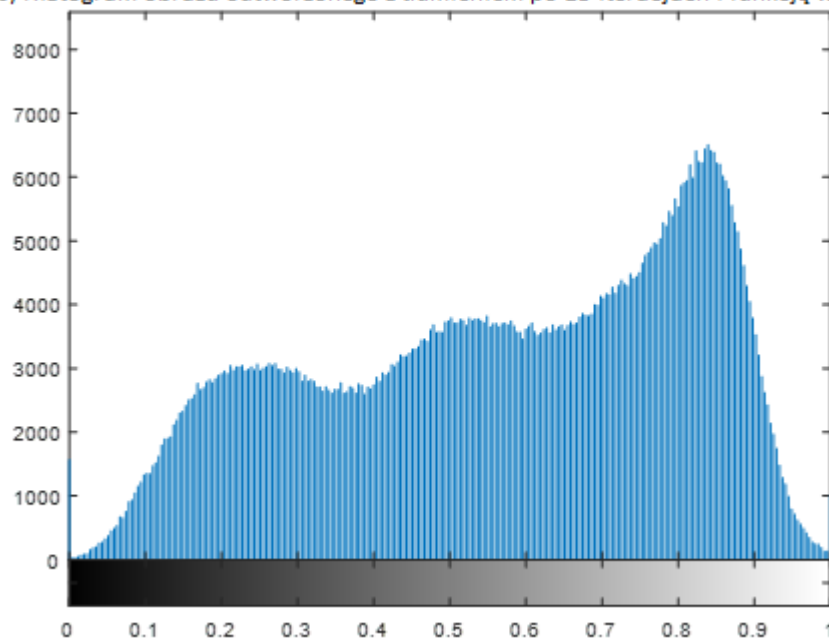


Rys 19. Rekonstrukcja obrazu po 15 iteracjach z tłumieniem wynoszącym $4 * \sqrt{v}$

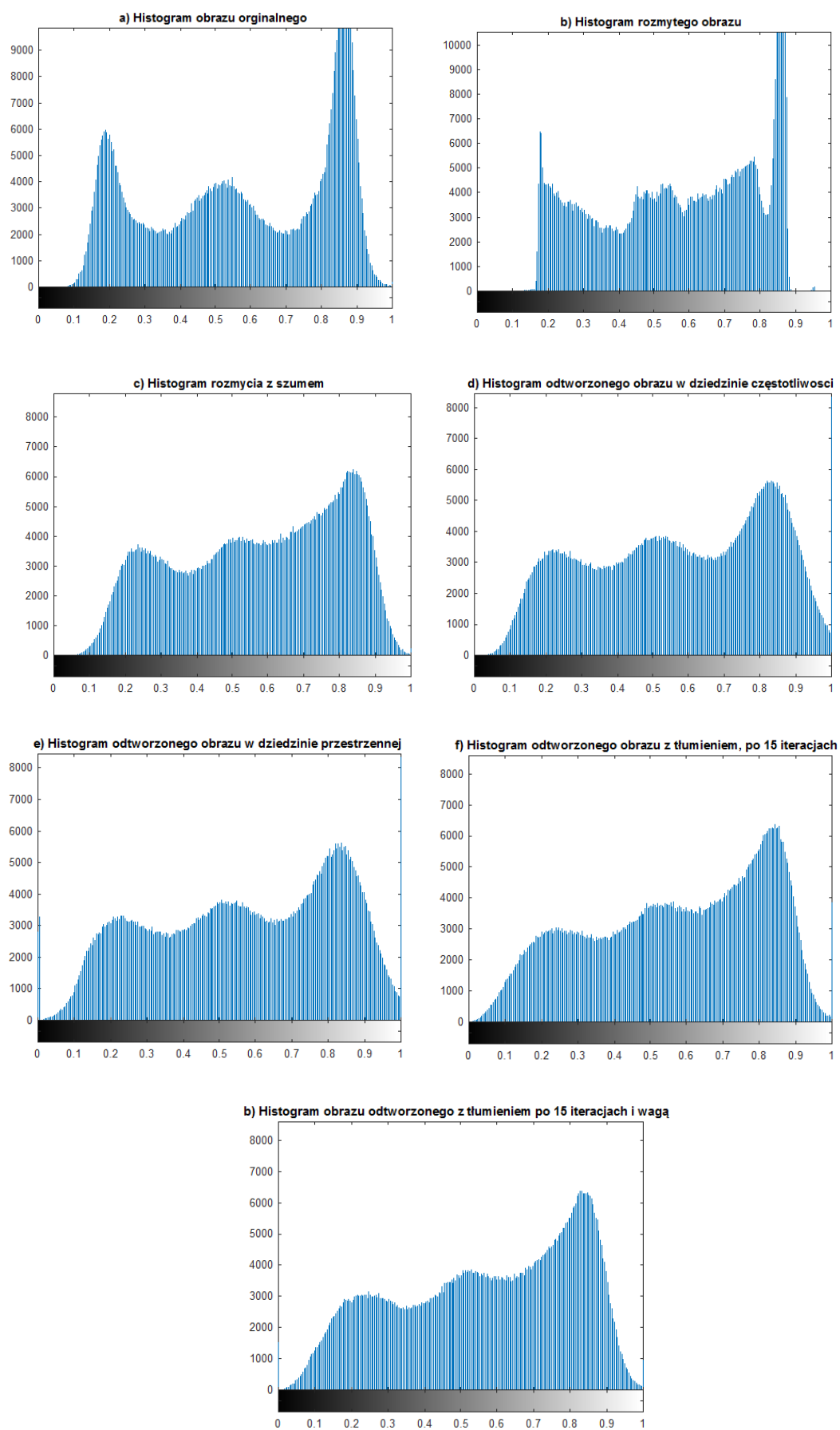
a) Odtworzony obraz



b) Histogram obrazu odtworzonego z tłumieniem po 15 iteracjach i funkcją ważoną



Rys 20. Rekonstrukcja obrazu po 15 iteracjach z tłumieniem i funkcją ważoną



Rys. 21. Zestawienie histogramów

8.3 Wnioski

Histogram obrazu rozmytego (rys. 21b) traci swoją gładkość względem histogramu obrazu oryginalnego (rys. 21a). Charakterystyczne 3 szczyty na histogramie obrazu oryginalnego zostały rozproszone i zniwelowane. Nakładając szum na rozmyty obraz można zaobserwować, że histogram rozmycia z szumem (rys. 21c) uległ wyrównaniu. Metodę Lucy-Richardsona zaimplementowano w dziedzinie przestrzennej (rys. 10) oraz w dziedzinie częstotliwościowej (rys. 12). Dla każdej implementacji zmierzono czas przetwarzania w Matlabie po 15 iteracjach. Rekonstrukcję w dziedzinie częstotliwości przedstawiono na rys. 17, zajmuje 1,839857 sekund i nie powoduje wyraźnej poprawy wizualnej względem rekonstrukcji przestrzennej. Rekonstrukcja przestrzenna (rys. 18) wynosi 0,572242 sekund, a więc jest prawie trzykrotnie krótsza czasowo od rekonstrukcji częstotliwościowej. Histogram odtworzonego obrazu w dziedzinie częstotliwości (rys. 21d) oraz histogram odtworzonego obrazu w dziedzinie przestrzennej (rys. 21e) są do siebie bardzo podobne. Na każdym widać już zarysy 3 szczytów histogramu oryginalnego. Ze względu na krótszy czas obliczeń, do dalszych rekonstrukcji wybrano odtworzony obraz w dziedzinie przestrzennej. Na wybrany obraz nałożono tłumienie szumu (rys. 18). Histogram odtworzonego obrazu z tłumieniem po 15 iteracjach (rys. 21f) ma podniesiony swój trzeci szczyt od lewej co odnosi się do wysokiego trzeciego szczytu na obrazie oryginalnym. Jednakże wysoki trzeci szczyt niweluje rozpoznawalność środkowego szczytu ze względu na zwiększoną liczbę pikseli dla argumentu 0,7. Ostatnim użytym parametrem w procesie rekonstrukcji była waga pikseli. Waga ma za zadanie niwelować powielanie się krawędzi na odtworzonym obrazie. Histogram odtworzonego obrazu z tłumieniem, po 15 iteracjach i funkcją ważoną (rys. 21g) nie różni się wyglądem od histogramu odtworzonego obrazu z tłumieniem po 15 iteracjach (rys. 21f). Różnica jest widoczna między obrazami na rys. 19 i rys. 20. Krawędzie powielające zostały wprawdzie zredukowane, ale w ich miejsce pojawiły się odbarwienia. Wynika to ze złych parametrów dla przyjętej funkcji ważonej.

9. Zastosowanie filtru Wienera do rekonstrukcji obrazu

Filtr Wienera jest użyteczny, gdy znana jest wiedza o rozmyciu i szumie. Działanie filtru opiera się na podejściu statystycznym, tzn. porównuje się obraz zaszumiony do obrazu oryginalnego. Filtr działa w dziedzinie częstotliwości, dlatego ważne jest, aby znać właściwości sygnału obrazu i amplitudę szumu.

Filtr Wienera należy do filtrów liniowych, więc opiera się na równaniu (7.1). W dziedzinie częstotliwości równanie (7.1) będzie miało następującą postać

$$G(u, v) = F(u, v)H(u, v) + N(u, v) \quad (9.1)$$

Do równania (9.1) przypisuje się filtr częstotliwościowy $Y(u, v)$, aby określić wartości jakościowe

$$\hat{F}(u, v) = Y(u, v)G(u, v) = Y(u, v)[F(u, v)H(u, v) + N(u, v)] \quad (9.2)$$

Filtr częstotliwościowy charakteryzuje się trzema właściwościami

- w przestrzeni częstotliwościowej, gdy składnik szumu $N(u, v)$ jest dużo mniejszy od składnika obrazu $G(u, v)$, filtr częstotliwościowy $Y(u, v)$ staje się odwrotnością funkcji rozmycia $H(u, v)$

$$Y(u, v) \approx \frac{1}{H(u, v)} \text{ dla } |N(u, v)| \ll |G(u, v)| \quad (9.3)$$

- gdy składnik szumu jest dużo większy od składnika obrazu, filtr przyjmuje wartość 0:

$$Y(u, v) \approx 0 \text{ dla } |N(u, v)| \gg |G(u, v)| \quad (9.4)$$

- dla częstotliwości, dla których wartości szumu i sygnału są porównywalne, filtr powinien znaleźć kompromis pomiędzy tolerancją szumu, a redukcją szumu.

Biorąc pod uwagę te trzy warunki filtr Wienera definiuje się równaniem:

$$G(u, v) = \frac{H^*(u, v)P_f(u, v)}{|H(u, v)|^2 P_f(u, v) + P_n(u, v)} \quad (9.5)$$

Podzielnie równania (9.5) przez $P_f(u, v)$ prowadzi do następującej postaci:

$$o(u, v) = \frac{H^*(u, v)}{|H(u, v)|^2 + \frac{P_n(u, v)}{P_f(u, v)}} \quad (9.6)$$

gdzie: $H^*(u, v)$ to spłot funkcji zniekształcającej z obrazem, $H(u, v)$ to funkcja zniekształcająca, $P_n(u, v)$ to moc szumu, $P_f(u, v)$ to moc sygnału obrazu oryginalnego, $\frac{P_n(u, v)}{P_f(u, v)}$ to stosunek mocy szumu do mocy sygnału NSR (noise-to-signal ratio).

9.1 Implementacja filtru Wienera

Pseudokod na rys. 22 przedstawia działanie filtru Wienera. Zaimplementowany filtr wykorzystuje przekształconą wersję równania (9.6). Równanie jest umieszczone w podwójnej pętli określającej rozmiar obrazu.

Algorithm 1 Filtr Wienera

```

1: %Określenia danych wejściowych zadania
2: obrazTF = FFT(obraz)
3: amplitudaSygnału ← fftshift( $\frac{\text{abs}(\text{obrazTF})^2}{\text{iloscPikseliObrazu}}$ )
4: szumTF = FFT(szum)
5: amplitudaSzumu ← fftshift( $\frac{\text{abs}(\text{szumTF})^2}{\text{iloscPikseliObrazu}}$ )
6: wartoscSzumu ← obliczenie wartosci szumu z amplitudy
7: wygladzenieZSzumem ← filtrowanie wygladzajace bazowego obrazu
8: wygladzenieZSzumemTF ← FFT(wygladzenieZSzumem);
9: amplituda ← fftshift( $\frac{\text{abs}(\text{wygladzenieZSzumemTF})^2}{(\text{iloscPikseliObrazu})}$ )
10: procedure WIENER(obraz)
11:   %Inicjalizacja danych
12:   filtrWypelniajacy ← wypełniono zerami, wielkosc rozmiaru obrazu
13:   filtrWypelniajacyFT ← wypełniono zerami, wielkość rozmiaru obrazu
14:   for u w zakresie rozmiar.x do
15:     for v w zakresie rozmiar.y do
16:       %Iteracja po wszystkich pikselach
17:        $NSR(u, v) \leftarrow \frac{\text{amplituda}(u, v)}{\text{amplituda}(u, v) + \text{wartoscSzumu}}$ 
18:       %Zastosowanie wzorow Wienera
19:        $\text{FiltrWypelniajacy} \leftarrow \frac{\text{zaszumionyObrazTF}(u, v)}{NSR(u, v)}$ 
20:     end for
21:   end for
22:   %Odwrocenie obrazu do bazowej przestrzeni
23:   OdwrconaFFT ← odwrconaFFT(FiltrWypelniajacy)
24:   WyjsciowyObraz ← czescRzeczywista(odwrconaFFT)
25: end procedure

```

Rys. 22. Pseudokod z implementacją filtru Wienera [11]

Implementacja filtru Wienera została wykonana w Matlabie (rys. 23).

```
%Rekonstrukcja filtrem Wienera w dziedzinie częstotliwości
Filtr_wypelniajacy_obraz=zeros(N);
Filtr_wypelniajacy_obrazFT=zeros(N);
for u=1:N
    for v=1:N
        %NSR (noise-to-signal response) stosunek mocy szumu do mocy sygnału
        NSR(u,v)=(Amplituda(u,v)+Wartosc_szumu)/Amplituda(u,v);
        Filtr_wypelniajacy_obrazFT(u,v)=Zaszumiony_obrazTF(u,v)/NSR(u,v);
    end
end

%odwrotna transformata Fouriera potrzebna by zwizualizować obraz
OdwroconaTF_dla_obrazu=(real(fft2(fftshift(Filtr_wypelniajacy_obrazFT))));

%konwertowanie na typ uint8
figure,imshow(uint8(OdwroconaTF_dla_obrazu)); |
title('a) Odtworzony obraz filtrem Wienera')
figure;
imhist(uint8(OdwroconaTF_dla_obrazu));
title('d) Histogram obrazu odtworzonego filtrem Wienera')
```

Rys. 23. Fragment kodu źródłowego z implementacją

9.2 Rekonstrukcja obrazu filtrem Wienera

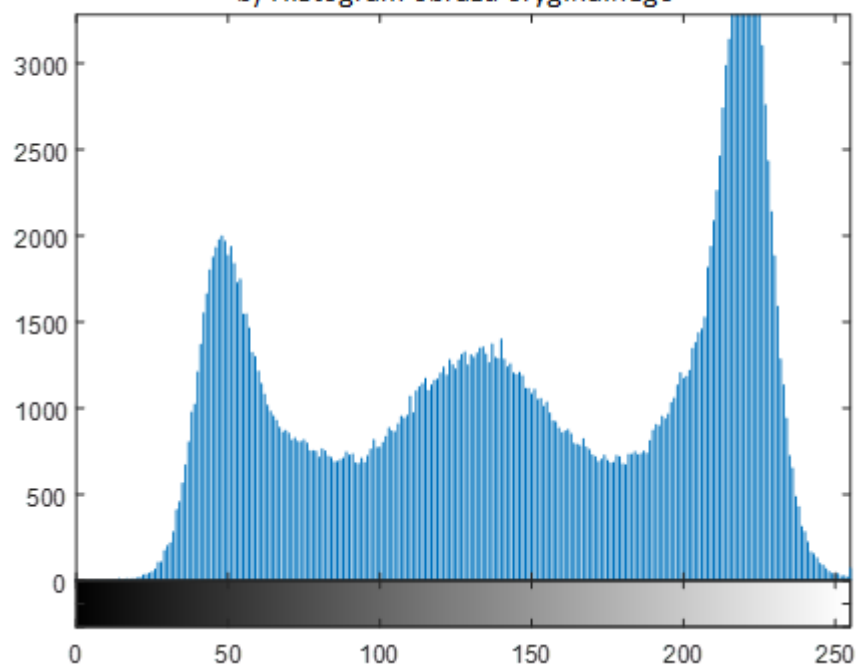
Rekonstrukcję rozpoczęto od wyznaczenia amplitudy sygnału obrazu oryginalnego o rozmiarze 512x512 (rys. 24). Następnie obraz został zaszumiony losową maską o odchyleniu standardowym równym 40 (rys. 25). Dla wykorzystanego szumu wyznaczono amplitudę w dziedzinie częstotliwości. Zaszumiony obraz również został przekształcony do dziedziny Fouriera. Następnie na zaszumiony obraz nałożono filtr gaussowski o rozmiarze 5x5 i odchyleniu standardowym 1. Filtr gaussowski pełni rolę filtru wygładzającego (rys 26). Zaszumiony i wygładzony obraz został przekształcony do dziedziny częstotliwości. Następnie została przeprowadzona rekonstrukcja zaimplementowaną metodą Wienera (rys. 22). Wykonane zostały jeszcze dwie rekonstrukcje w celu zbadania wpływu NSR (stosunek mocy szumu do mocy sygnału) na jakość odtworzonego obrazu. Pierwsza rekonstrukcja została wykonana dla NSR wymnożonego przez 1,5. Druga rekonstrukcja została wykonana dla NSR wymnożonego przez 0,5.

a) Obraz oryginalny



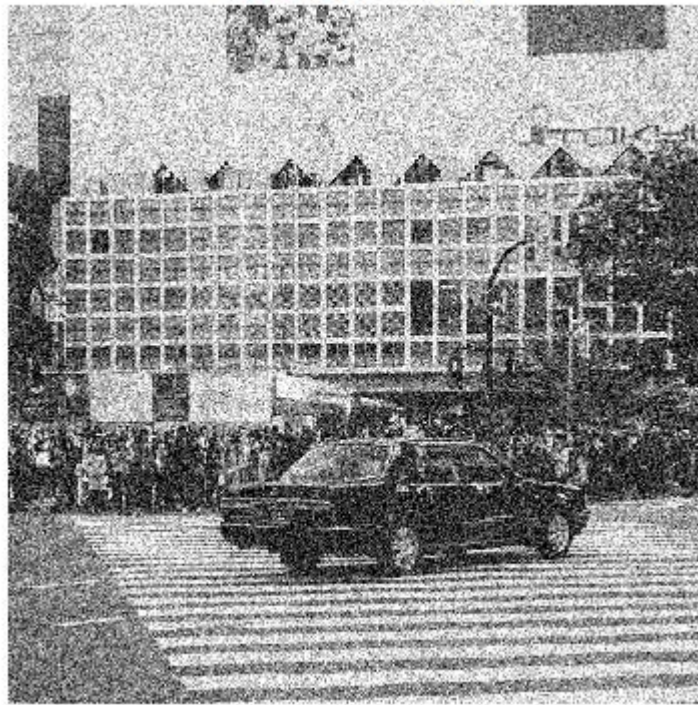
Photo by Jon Tang on Unsplash, link - <https://unsplash.com/photos/UD0YE6DnU>

b) Histogram obrazu oryginalnego

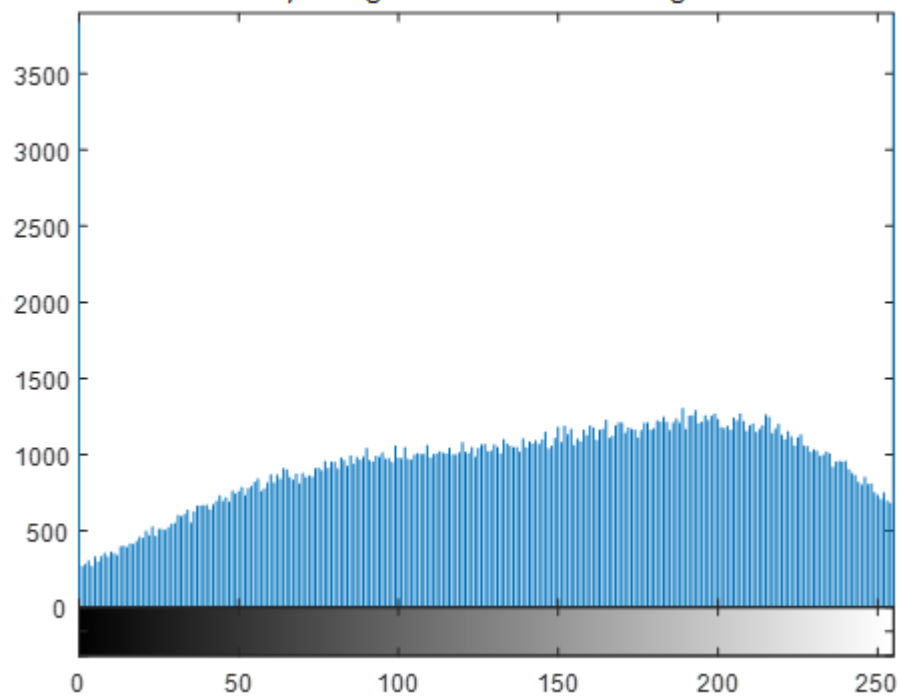


Rys. 24. Obraz oryginalny i jego histogram

a) Obraz zaszumiony



b) Histogram obrazu zaszumionego

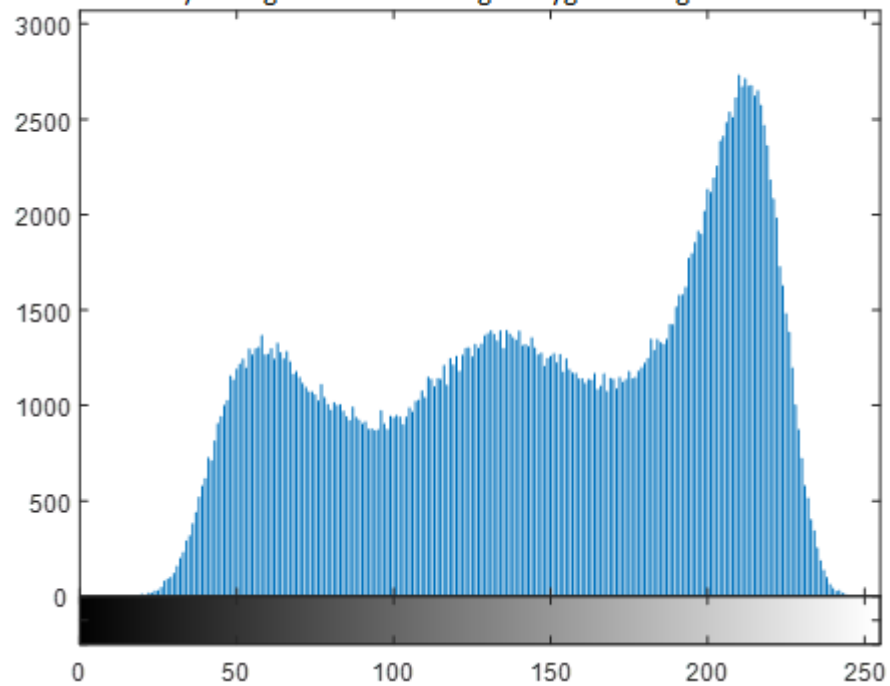


Rys. 25. Zaszumiony obraz i jego histogram

a) Zaszumiony obraz po wygładzeniu filtrem gaussowskim



b) Histogram zaszumionego i wygładzonego obrazu

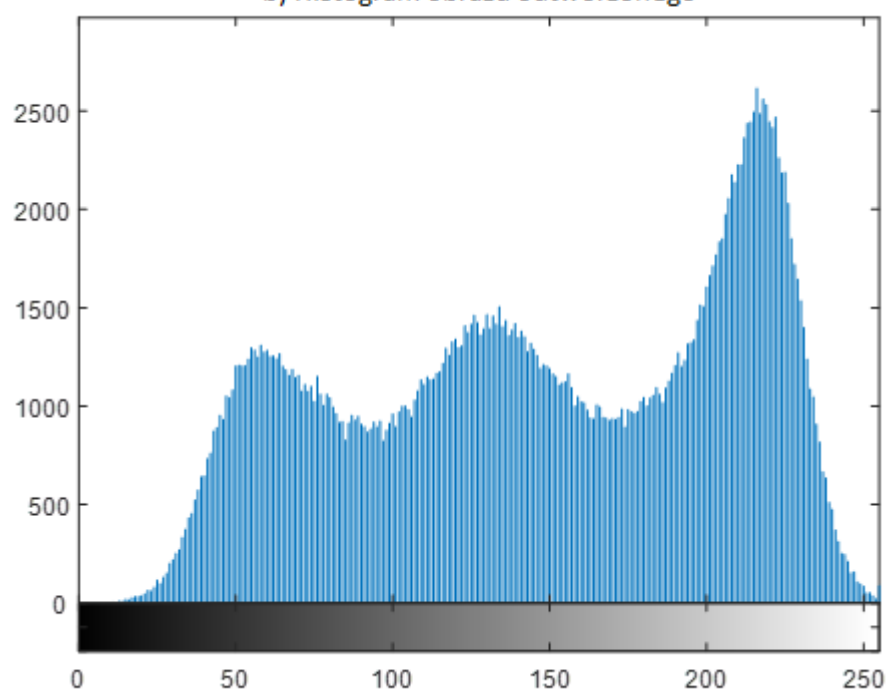


Rys. 26. Zaszumiony i wygładzony obraz oraz jego histogram

a) Odtworzony obraz filtrem Wienera



b) Histogram obrazu odtworzonego

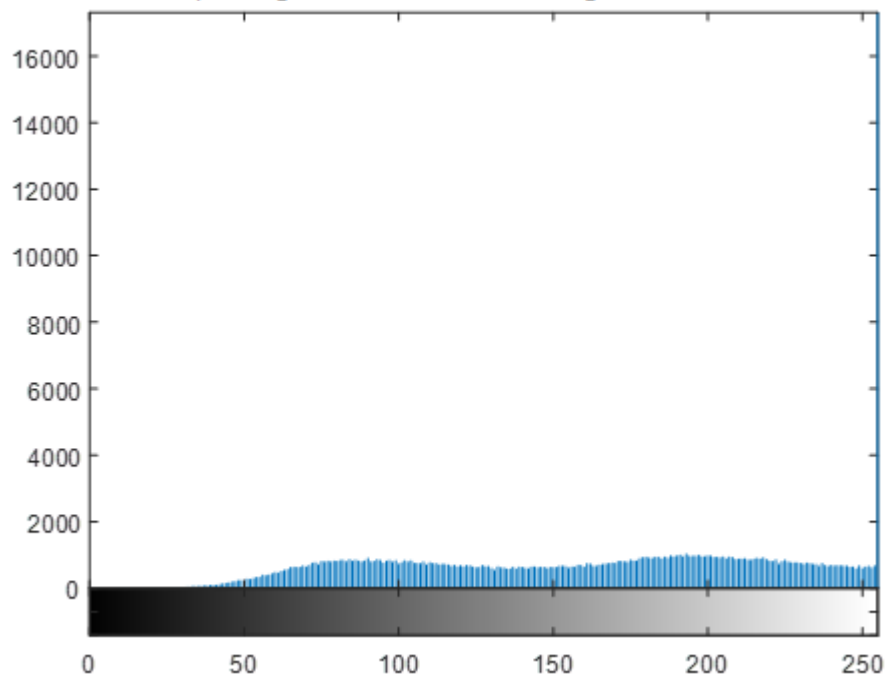


Rys. 27. Rekonstrukcja obrazu filtrem Wienera

a) Odtworzony obraz filtrem Wienera dla $NSR \cdot 1,5$



b) Histogram obrazu odtworzonego dla $NSR \cdot 1,5$

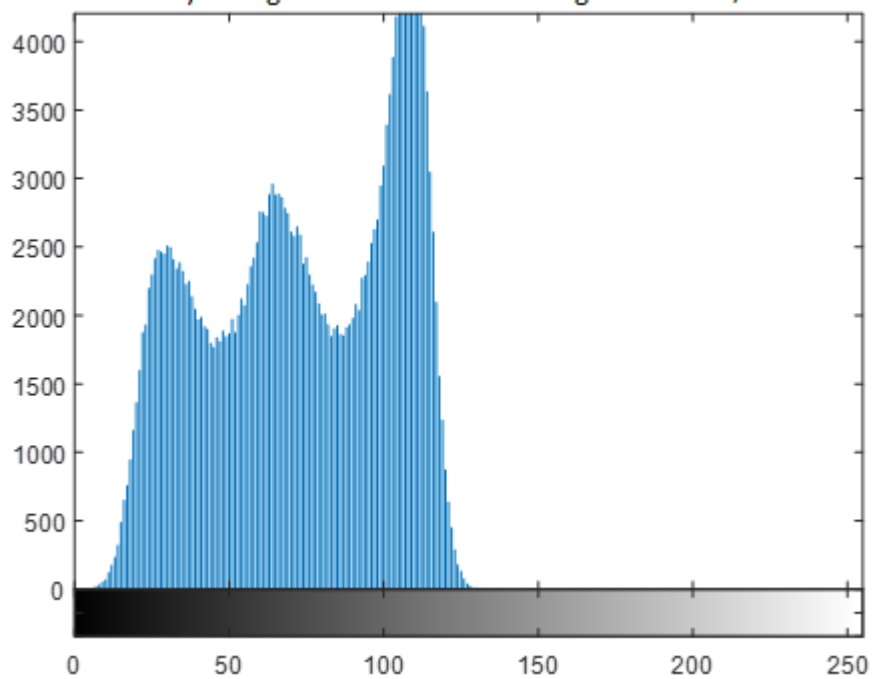


Rys. 28. Odtworzony obraz filtrem Wienera oraz jego histogram, dla NSR wymnożonego przez 1,5

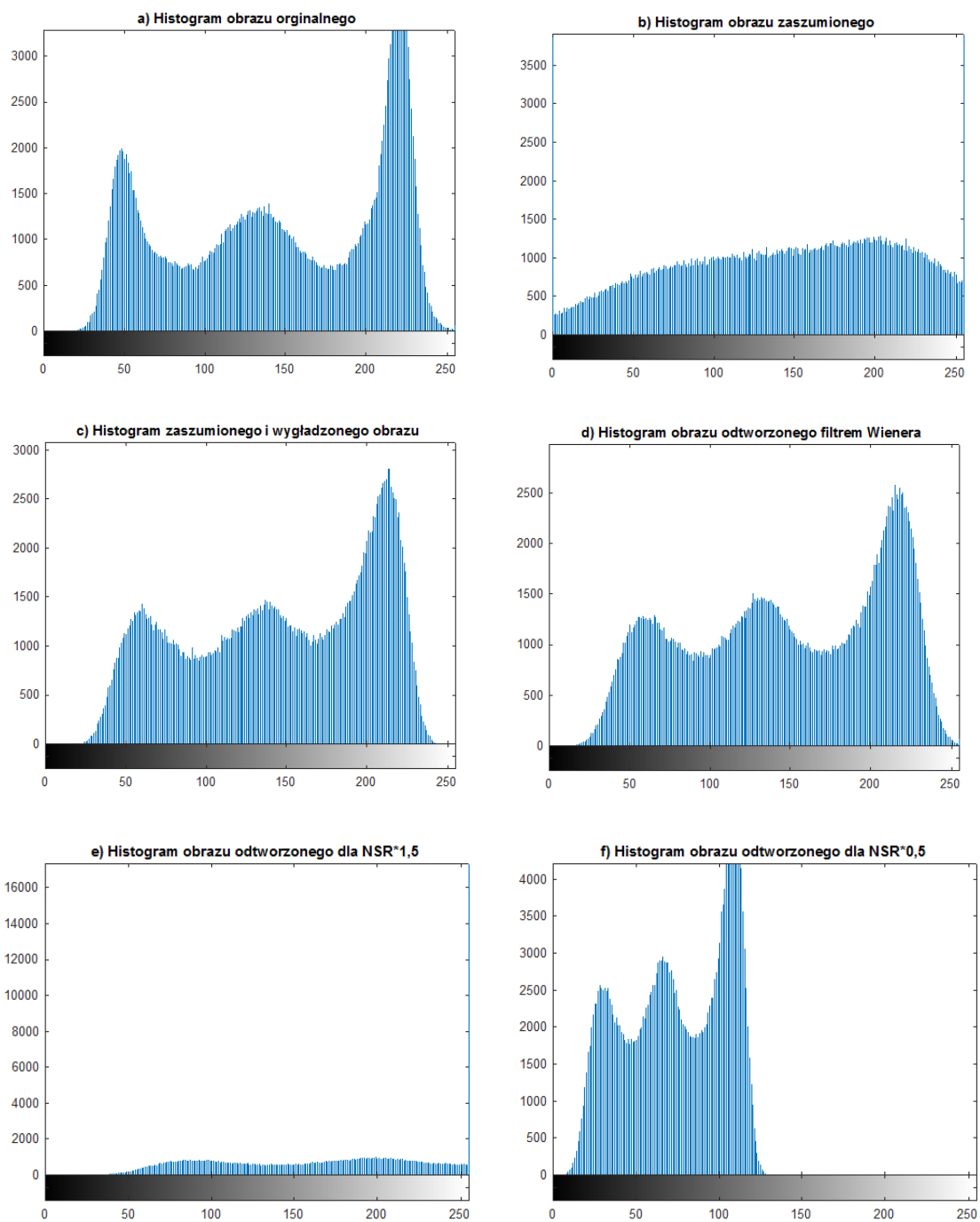
a) Odtworzony obraz filtrem Wienera dla $NSR \cdot 0,5$



b) Histogram obrazu odtworzonego dla $NSR \cdot 0,5$



Rys. 29. Odtworzony obraz filtrem Wienera oraz jego histogram, dla NSR wymnożonego przez 0,5



Rys. 30 Zestawienie histogramów

9.3 Wnioski

Zaimplementowany szum spowodował degradację obrazu (rys. 25). Histogram obrazu zaszumionego (rys. 30b) uległ obniżeniu, a liczba pikseli odpowiadających argumentom w skali szarości została prawie wyrównana. Największa liczba pikseli w histogramie obrazu zaszumionego przypada dla wartości 0 i 256. Wygładzenie filtrem gaussowskim spowodowało poprawę obrazu (rys. 26). Szum został zniwelowany, poprawie uległ też kontrast. Histogram zaszumionego i wygładzonego obrazu (rys. 26) jest podobny do histogramu obrazu oryginalnego. Trzy szczyty znajdujące się na obrazie oryginalnym zostały również odwzorowane na obrazie wygładzonym. Wygładzony i zaszumiony obraz poddano rekonstrukcji zaimplementowanym filtrem Wienera. Odtworzony obraz (rys. 27) nie przedstawia poprawy względem obrazu wygładzonego (rys. 26), którą można zauważyć gołym okiem. Histogram obrazu odtworzonego filtrem Wienera (rys. 30d) jest jednak lepszy od histogramu zaszumionego i wygładzonego obrazu (rys. 30c). Na histogramie obrazu odtworzonego filtrem Wienera środkowe maksimum oraz dwa sąsiadujące mu minima lokalne mają zaokrąglony kształt. Na histogramie wygładzonym środkowy szczyt i rowki są chropowate. Rekonstrukcja dla NSR wymnożonego przez 1,5 powoduje rozjaśnienie obrazu (rys. 28). Histogram obrazu odtworzonego dla $NSR \cdot 1,5$ (rys. 30e) jest bardzo płaski, dominuje jasność, dlatego największa liczba pikseli jest przypisana do wartości 256 w skali jasności. Rekonstrukcja dla NSR wymnożonego przez 0,5 powoduje przyciemnienie obrazu (rys. 29). Histogram obrazu odtworzonego dla $NSR \cdot 0,5$ (rys. 30f) przyjmuje wartości tylko od 0 do 125 w skali szarości.

10. Podsumowanie

Rekonstrukcja obrazów pojawia się w wielu dziedzinach techniki i przez to jest powszechnie stosowana w wielu aspektach życia codziennego. Rekonstrukcja obrazów należy do szerokiego grona zagadnień odwrotnych. W pracy przedstawiono numeryczne metody oparte o algorytmy i filtry, których zadaniem jest przywrócenie zniekształconego obrazu, a przez to rozwiązanie problemu odwrotnego.

Poprawność rekonstrukcji obrazu w numerycznych metodach rozwiązywania problemu odwrotnego zależy w dużym stopniu od prawidłowego zdefiniowania zagadnienia i doborze warunków początkowych. Błędy numeryczne wpływające na jakość odtworzonego obrazu i szukanych parametrów to m. in. zła klasyfikacja obrazu, niepoprawna implementacja algorytmu czy też niewystarczająca ilość iteracji.

W części projektowej zdefiniowano i przeprowadzono rekonstrukcję z wykorzystaniem algorytmu Lucy-Richardsona oraz filtru Wienera. Najlepszy rezultat wizualny osiągnięto dla rekonstrukcji filtrem Wienera o niezmodyfikowanym stosunku amplitudy szumu do amplitudy sygnału. Jak się okazało, w tym przypadku najlepsza jakość obrazu nie odbiła się na idealnym odwzorowaniu histogramu obrazu odtworzonego do histogramu obrazu oryginalnego, niemniej jednak rezultat jest zadowalający. Algorytm Lucy-Richardsona wykorzystano do przeprowadzenia dwóch rekonstrukcji. Pierwsza została wykonana w dziedzinie częstotliwości, a druga w dziedzinie przestrzennej. Otrzymane obrazy były bardzo podobne pod względem jakości, jednakże bardzo różniły się pod względem czasu wykonywania operacji.

Rozwiązywanie problemu odwrotnego w rekonstrukcji obrazu przynosi lepsze bądź gorsze efekty w zależności od wybranej metody. Użyte metody mogą być rozwijane przez dodanie kolejnych parametrów, a efektywność parametrów zależy od zastosowanych wartości liczbowych. Dobór parametrów i ich wartości następuje często za pomocą metody prób i błędów, aż do uzyskania najlepszego rezultatu.

Bibliografia

- [1] Iwaniec J.: *Wybrane zagadnienia eksploatacyjnej identyfikacji układów nieliniowych*. Kraków 2011
- [2] Bernd J.: *Digital Image Processing*. Springer-Verlag Berlin Heidelberg 2005
- [3] Solomon C., Breckon T.: *Fundamentals of Digital Image Processing A Practical Approach with Examples in Matlab*. John Wiley & Sons, Ltd 2011
- [4] Jain A. K.: *Fundamentals of Digital Image Processing*. New Jersey Prentice-Hall, Inc 1989
- [5] Tadeusiewicz R., Korohoda P.: *Komputerowa analiza i przetwarzanie obrazów*. Kraków, Wydawnictwo Fundacji Postępu Telekomunikacji 1997
- [6] Tarantola A.: *Inverse Problem Theory and Methods for Model Parameter Estimation*. Filadelfia, Society for Industrial and Applied Mathematics 2005
- [7] Hwan K. J., McCann M. T., Froustey E., Unser M., *Deep Convolutional Neural Network for Inverse Problems in Imaging*. IEEE Transactions on Image Processing, vol, 26. No. 9, September 2017
- [8] Autor zdjęcia Jon Tang, Dostęp: <https://unsplash.com/photos/-JODYEU6DmU>
- [9] Autor zdjęcia Argan Yoga Nugroho Dostęp: <https://images.unsplash.com/photo-1546530728-628d62351932?ixlib=rb-1.2.1&q=85&fm=jpg&crop=entropy&cs=srgb&dl=argan-yoga-nugroho-1269714-unsplash.jpg>
- [10] <http://www.algorytm.org/przetwarzanie-obrazow/filtrowanie-obrazow.html>
- [11] <https://www.overleaf.com/project>

Załączniki:

1. Treść kodu z programu Matlab z algorytmem Lucy-Richardsona

```
%Metoda z użyciem algorytmu Lucy-Richardsona do deblurowania obrazów.
%Może być efektywnie stosowana, gdy znana jest funkcja rozmycia
punktów FRP (operator rozmycia), ale informacja o szumie jest
niewielka lub nie jest dostępna.
%Zamazany i zaszumiony obraz zostaje przywrócony przez iteracyjny,
przyspieszony, tłumiony algorytm Lucy-Richardsona.
%Można użyć charakterystyki układu optycznego jako parametrów
wejściowych, aby poprawić jakość przywracania obrazu.

%Funkcja dekonwolucji może obsługiwać tablice o dowolnym rozmiarz
clc
clear all

I = im2double(imread('jon-tang490.jpg'));

figure;
imshow(I);
title('Obraz oryginalny');
text(size(I,2),size(I,1)+15, ...
    'Photo by Jon Tang on Unsplash, link -
    https://unsplash.com/photos/-JODYEU6DmU', ...
    'FontSize',7,'HorizontalAlignment','right');
figure;
imhist(I);title('Histogram obrazu oryginalnego')

rozmiar=[5 5]; sigma=5;
FRP = fspecial('gaussian', rozmiar, sigma);
Rozmycie = imfilter(I, FRP);
figure; imshow(Rozmycie); title('Obraz rozmyty')
figure; imhist(Rozmycie); title('Histogram rozmytego obrazu ')

Srednia_szumu = 0; %wpływa na jasność
Amplituda_szumu = 0.002;
V = .002;
Rozmycie_z_szumem = imnoise(Rozmycie,'gaussian',0,V);
figure;
imshow(Rozmycie_z_szumem);
title('Rozmycie z szumem na obrazie początkowym');
figure;
imhist(Rozmycie_z_szumem);
title('Histogram rozmycia z szumem ');

tic;
wynikRLF = czestotRL(Rozmycie_z_szumem, FRP, 15); toc;
figure; imshow(wynikRLF); title('Odtworzony obraz w dziedzinie
częstotliwosci')
figure; imhist(wynikRLF); title('Histogram odtworzonego obrazu w
dziedzinie częstotliwosci')

tic;
wynikRL = dekonwolucjaRL(Rozmycie_z_szumem, FRP, 15); toc;
figure; imshow(wynikRL); title('Odtworzony obraz')
figure; imhist(wynikRL); title('Histogram odtworzonego obrazu')

%Kontrola wzmocnienia szumu poprzez tłumienie
```

```
%Najnowszy obraz, LucyRich2, jest wynikiem 15 iteracji.
%Chociaż jest ostrzejszy niż poprzedni wynik z 5 iteracji, obraz
rozwiija "plamkowaty" wygląd.
%Plamki nie odpowiadają żadnym prawdziwym strukturom (co widać poprzez
porównanie z pierwotnym obrazem), ale są wynikiem zbytniego
dopasowania szumu do danych.
```

```
Tlumik = im2double(4*sqrt(V));
LucyRich3 = deconvlucy(Rozmycie_z_szumem,FRP,15,Tlumik);
figure;
imshow(LucyRich3);
title('Odtworzony obraz z tłumieniem, po 15 iteracjach');
figure;
imhist(LucyRich3)
title('Histogram odtworzonego obrazu z tłumieniem, po 15 iteracjach');
```

```
%Zostanie teraz wykonana rekonstrukcja z funkcją Waga, która działa na
samym rozmyciu obrazu.
%Tworzymy macierz funkcji ważenia, Waga, która składa się z jedynek w
środkowej części zamazanego obrazu ("dobre" piksele, znajdujące się w
linii przerywanej) i zer na krawędziach ("złe" piksele - te, które nie
odbierają sygnału ).
```

```
%Wykonano macierz Waga zgodną z romiarem obrazu oryginalnego
%Algorytm waży każdą wartość piksela zgodnie z tablicą Waga podczas
przywracania obrazu.
%W poniższym przypadku używane są tylko wartości środkowych pikseli
(gdzie WEIGHT = 1), podczas gdy "złe" wartości pikseli są wykluczone z
optymalizacji.
```

```
K = im2double(Rozmycie_z_szumem)
Waga = zeros(size(I));
Waga(5:end-4,5:end-4) = 1;
CutImage = K.*Waga;
CutEdged = edgetaper(CutImage,FRP);
DAMPAR = im2double(4*sqrt(V));
LucyRich4 = deconvlucy(CutEdged,FRP,15,DAMPAR,Waga);
figure; imshow(LucyRich4)
title('Oczyszczony obraz')
figure; imhist(LucyRich4);
title('Histogram obrazu oczyszczonego z funkcja ważona')
```

```
function result = dekonwolucjaRL(obraz, FRP, iter)
    % obrazy muszą być w formacie double by wykonać konwolucję
    obraz = double(obraz);
    FRP = double(FRP);
    obrazdys = obraz; % wprowadzenie na obraz dyskretny
    OdwrFRP = FRP(end:-1:1,end:-1:1); % przestrzennie odwrócona PSF
    dla obrazu dyskretnego
```

```
    for i= 1:iter
        konwolucja = convn(obrazdys,FRP,'same');
        Rozmycie2 = obraz./konwolucja;
        blad = convn(Rozmycie2,OdwrFRP,'same');
        obrazdys = obrazdys.* blad;
    end
```

```
    result = obrazdys;
```



```

end

%funkcja w dziedzinie czestotliwosci

function result = czestotRL(obraz, FRP, iter)
obraziter = obraz; %wczytanie obrazu do iteracji
OFP = psf2otf(FRP,size(obraz)); %OFP optyczna funkcja przenoszenia
%z ang. OTF-optical transfer function, zmiana FRP na OTF

for i=1:iter
    uklad = fft2(obraziter); %przekształcenie do dziedziny
    częstotliwości
    splot = OFP.*uklad;
    transformacjaodw = ifft2(splot); %odwrotna transformacja
    %fouriera(czestotliwosciowa)
    stosunek = obraz./transformacjaodw;
    ukladstosunku = fft2(stosunek);
    wynik = OFP .* ukladstosunku;
    transfwyniku = ifft2(wynik);
    obraziter = transfwyniku.*obraziter;
end
result = abs(obraziter);
end

```

2. Treść kodu z programu Matlab z implementacją filtru Wienera.

```
% w poniższej metodzie wykorzystano filtr Wienera do usunięcia
zakłóceń.
% Metoda ta może być zastosowana, gdy funkcja rozmycia punktu i poziom
% zaszumienia jest znany
clc
clear all

%wczytanie obrazu
K=imread('jon-tang512.jpg');
I=rgb2gray(K);
imshow(I);
title('a) Obraz oryginalny');
N=512;
%zmiana formatu obrazu na double
Obraz=double(I);
text(size(I,2),size(I,1)+15, ...
    'Photo by Jon Tang on Unsplash, link -
https://unsplash.com/photos/-JODYEU6DmU', ...
    'FontSize',7,'HorizontalAlignment','right');
figure;
imhist(I);title('a) Histogram obrazu oryginalnego')

%Wyznaczenie amplitudy sygnału dla obrazu oryginalnego
ObrazTF=fft2(Obraz); %przekształcenie obrazu do dziedzi-
ny częstotliwości (transformacja Fouriera TF)
Amplituda_sygnalu=fftshift((abs(ObrazTF).^2)./(N*N));

%Nałożenie szumu na obraz o sigmie równej 40
sigma=40;
Szum=sigma*randn(size(Obraz));
figure;
imshow(uint8(Obraz+Szum));
title('a) Obraz zaszumiony');
figure;
imhist(uint8(Obraz+Szum));
title('b) Histogram obrazu zaszumionego')

%Wyznaczenie amplitudy szumu
SzumTF=fft2(Szum); % przekształcenie do dziedzi-
ny częstotliwości (transformacja Fouriera TF)
Amplituda_szumu=fftshift((abs(SzumTF).^2)./(N*N));
Wartosc_szumu=sum(sum(Amplituda_szumu))/(N*N)

%Transformacja Fouriera zaszumionego obrazu
L=Obraz;
L=L+Szum;
LFT=fft2(L);
Zaszumiony_obrazTF=fftshift(LFT);

%Nałożenie filtru gaussowskiego na zaszumiany obraz
%Filtr gaussowski działa w tym wypadku jako filtr wygładzający
Filtr_wygładzający = fspecial('gaussian',[5 5], 1);
Wygładzenie_z_szumem = imfilter(uint8(L), Filtr_wygładzający,
    'replicate');
figure,imshow(Wygładzenie_z_szumem);
title('a) Zaszumiony obraz po wygładzeniu filtrem gaussowskim');
```

```

figure;
imhist(Wygladzenie_z_szumem);
title('c) Histogram zaszumionego i wygładzonego obrazu')

%Amplituda sygnału dla zaszumionego i wygładzonego obrazu
Wygladzenie_z_szumemTF=fft2(Wygladzenie_z_szumem); % przekształcenie
do dziedziny częstotliwości (transformacja Fouriera TF)
Amplituda=fftshift((abs(Wygladzenie_z_szumemTF).^2)./(N*N));

%Rekonstrukcja filtrem Wienera w dziedzinie częstotliwości
Filtr_wypelniajacy_obraz=zeros(N);
Filtr_wypelniajacy_obrazFT=zeros(N);
for u=1:N
    for v=1:N
        %NSR (noise-to-signal response) stosunek mocy szumu do mocy
        sygnału
        NSR(u,v)=(Amplituda(u,v)+Wartosc_szumu)/(Amplituda(u,v));

Filtr_wypelniajacy_obrazFT(u,v)=Zaszumiony_obrazTF(u,v)/NSR(u,v);
    end
end

%odwrotna transformata Fouriera potrzebna by zwizualizować obraz
OdwroconaTF_dla_obrazu=(real(ifft2(ifftshift(Filtr_wypelniajacy_obrazF
T))));

%konwertowanie na typ uint8
figure,imshow(uint8(OdwroconaTF_dla_obrazu));
title('a) Odtworzony obraz filtrem Wienera')
figure;
imhist(uint8(OdwroconaTF_dla_obrazu));
title('d) Histogram obrazu odtworzonego filtrem Wienera')

%Rekonstrukcja dla NSR równego 1.5
Filtr_wypelniajacy_obraz=zeros(N);
Filtr_wypelniajacy_obrazFT=zeros(N);
for u=1:N
    for v=1:N
        %NSR (noise-to-signal response) stosunek mocy szumu do mocy
        sygnału

NSR15(u,v)=1.5*(Amplituda(u,v)/(Amplituda(u,v)+Wartosc_szumu));

Filtr_wypelniajacy_obrazFT(u,v)=Zaszumiony_obrazTF(u,v)*NSR15(u,v);
    end
end

%odwrotna transformata Fouriera potrzebna by zwizualizować obraz
OdwroconaTF_dla_obrazu=(real(ifft2(ifftshift(Filtr_wypelniajacy_obrazF
T))));

%konwertowanie na typ uint8
figure,imshow(uint8(OdwroconaTF_dla_obrazu));
title('a) Odtworzony obraz filtrem Wienera dla NSR*1,5')
figure;
imhist(uint8(OdwroconaTF_dla_obrazu));

```

```

title('e) Histogram obrazu odtworzonego dla NSR*1,5')

%Rekonstrukcja dla NSR równego 0.5
Filtr_wypelniajacy_obraz=zeros(N);
Filtr_wypelniajacy_obrazFT=zeros(N);
for u=1:N
    for v=1:N
        %NSR (noise-to-signal response) stosunek mocy szumu do mocy
        sygnału

        NSR05(u,v)=0.5*(Amplituda(u,v)/(Amplituda(u,v)+Wartosc_szumu));

        Filtr_wypelniajacy_obrazFT(u,v)=Zaszumiony_obrazTF(u,v)*NSR05(u,v);
    end
end

%odwrotna transformata Fouriera potrzebna by zwizualizować obraz
OdwroconaTF_dla_obrazu=(real(ifft2(ifftshift(Filtr_wypelniajacy_obrazF
T))));

%konwertowanie na typ uint8
figure,imshow(uint8(OdwroconaTF_dla_obrazu));
title('a) Odtworzony obraz filtrem Wienera dla NSR*0,5')
figure;
imhist(uint8(OdwroconaTF_dla_obrazu));
title('f) Histogram obrazu odtworzonego dla NSR*0,5')

```