

**Wyższa Szkoła Informatyki Stosowanej i Zarządzania  
WIT**

**Grupa IZ07IO1**

**ALGORYTMY PRZETWARZANIA OBRAZÓW  
Aplikacja zbiorcza (ćwiczenia laboratoryjne i projekt)**

**Tytuł projektu  
Program interaktywnej konwersji obrazów**

**Wykonali:**  
Piotr Martycz-Jaworski  
Damian Patoka

**Prowadzący: dr inż. Marek Doros**  
Warszawa 2016/2017

<b>ALGORYTMY PRZETWARZANIA OBRAZÓW</b>	<b>1</b>
<b>Przygotowana aplikacja</b>	<b>3</b>
<b>Laboratorium 1: Wyrównywanie histogramu</b>	<b>4</b>
1. Metoda średnich	4
2. Metoda losowa	4
3. Metoda sąsiedztwa	5
4. Metoda własna	5
<b>Laboratorium 2: Operacje punktowe</b>	<b>6</b>
1. Operacje punktowe jednoargumentowe	6
Negacja	6
Progowanie	6
Redukcja poziomów szarości	8
Rozciąganie	9
2. Uniwersalny Operator Punktowy	10
3. Operacje punktowe dwuargumentowe arytmetyczne i logiczne	11
Arytmetyczna suma	11
Arytmetyczna różnica	12
Arytmetyczny iloczyn	12
Logiczne AND	13
Logiczne OR	13
Logiczne XOR	14
<b>Laboratorium 3: Operacje sąsiedztwa</b>	<b>15</b>
1. Filtracja liniowa	15
Wygładzenie	15
Wyostrowanie	16
Detekcja krawędzi	17
2. Filtry statystyczne	19
Minimalny	19
Maksymalny	19
Medianowy	19
<b>Laboratorium 4: Operacje sąsiedztwa</b>	<b>21</b>
1. Ścienianie	21
2. Operacje morfologiczne erozji, dylatacji, otwarcia, zamknięcia	21
Erozja	21
Dylatacja	22
Otwarcie	22
Zamknięcie	23
<b>Laboratorium 5: algorytmy segmentacji i analizy obrazu</b>	<b>24</b>
1. Algorytm żółwia	24
<b>Projekt: Konwersja obrazów szarych na kolorowe</b>	<b>25</b>
1. Omówienie	25
2. Algorytm koloryzacji	28
<b>Literatura</b>	<b>29</b>

## Przygotowana aplikacja

Aplikacja została stworzona w technologii HTML i JavaScript. Można z niej korzystać za pomocą przeglądarek internetowych, najlepiej Chrome.

Odpowiednie opcje menu pozwalają otworzyć jeden lub więcej plików graficznych, wybrać rodzaj transformacji, resetować zmiany transformacji. Po wyborze transformacji często pojawiają się dodatkowe kontrolki, którymi można wybrać tryb i zakres danego przekształcenia obrazu.

Obraz prezentowany jest w dwóch sekcjach: Oryginał i Modyfikacje. Obie sekcje zawierają histogram. Obraz po otwarciu jest konwertowany do skali szarości, przygotowane transformacje również działają w tym zakresie.

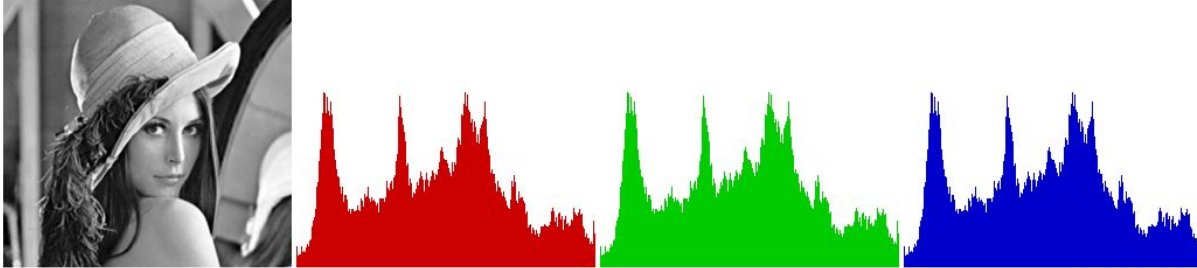
Aplikacja zawiera również projekt. Jego zadaniem jest przekształcenie obrazów szarych na kolorowe.

# Laboratorium 1: Wyrównywanie histogramu

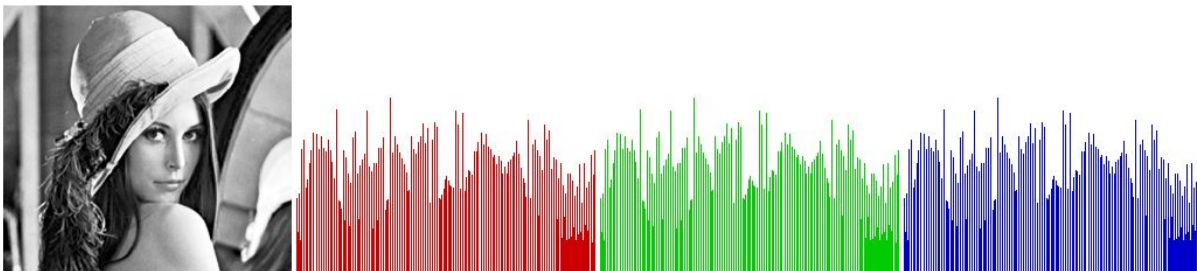
## 1. Metoda średnich

Metoda nie daje najlepszych z możliwych rezultatów, ale jest najprostsza i najszybciej się wykonuje. Każdy poziom jasności jest odwzorowywany w wartość średnią całego przedziału

Oryginał



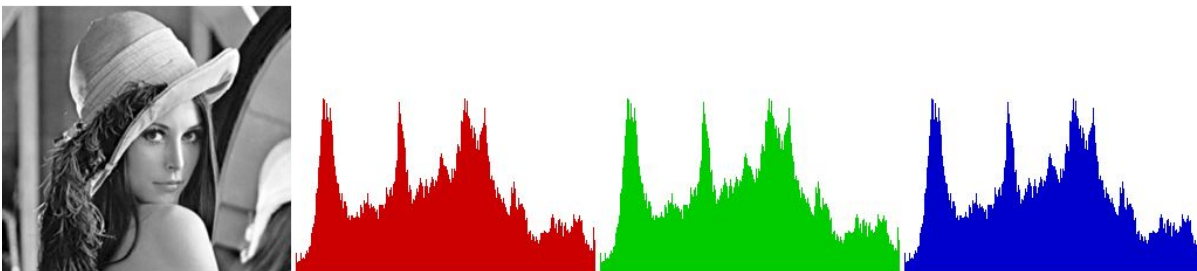
Modyfikacje



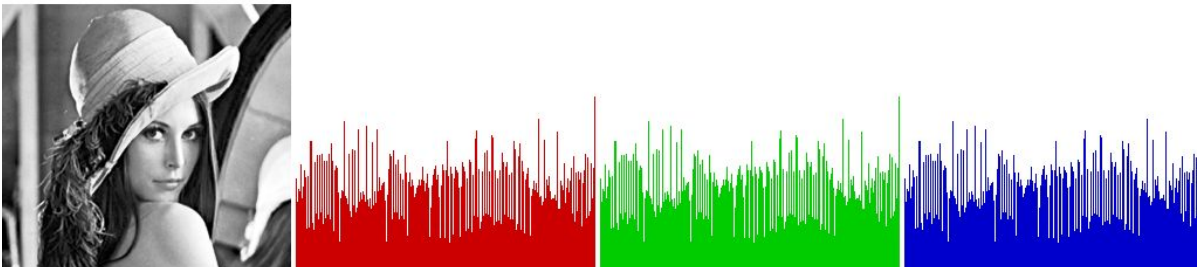
## 2. Metoda losowa

Jest bardziej złożona obliczeniowo od metody średnich. Pozwala zapobiegać błędom statystycznym. Może powodować utratę kontrastu, jeśli histogram pierwotny ma oddalone od siebie dwa wyróżnione szczyty. Metoda polega na odwzorowaniu poziomów jasności na losowe wartości z danego przedziału.

Oryginał



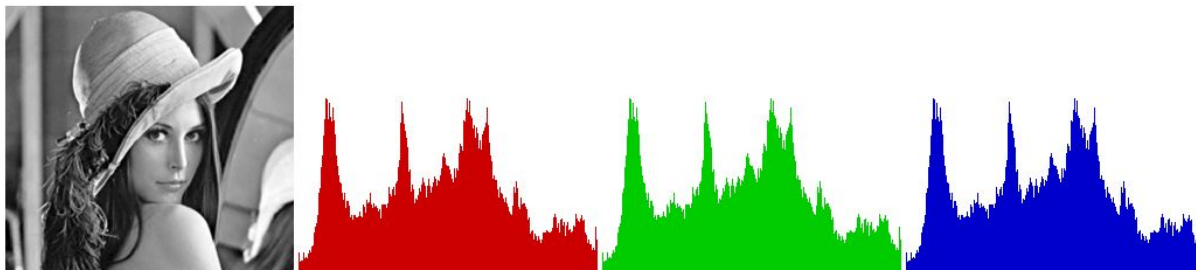
Modyfikacje



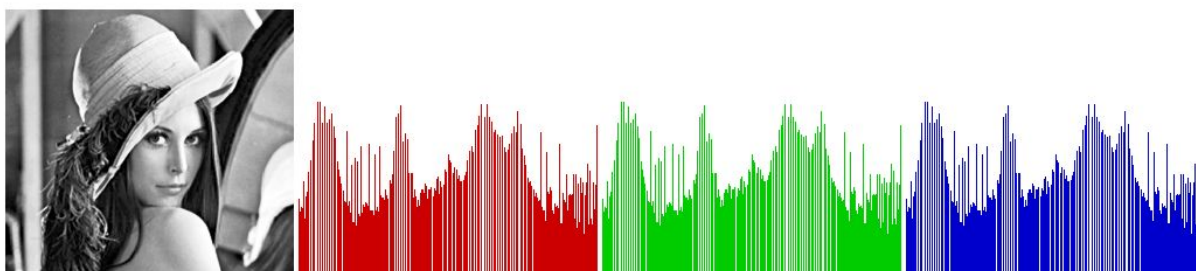
### 3. Metoda sąsiedztwa

Jest bardziej złożona obliczeniowo od metody średnich i losowej. Daje bardziej dokładne rezultaty. U podstaw działania metody leży założenie, że elementy będą mieć tym bardziej zbliżone wartości im bliżej siebie leżą.

Oryginał



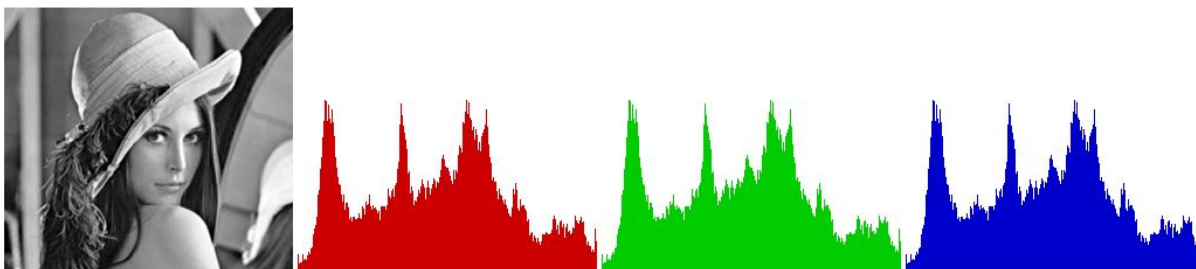
Modyfikacje



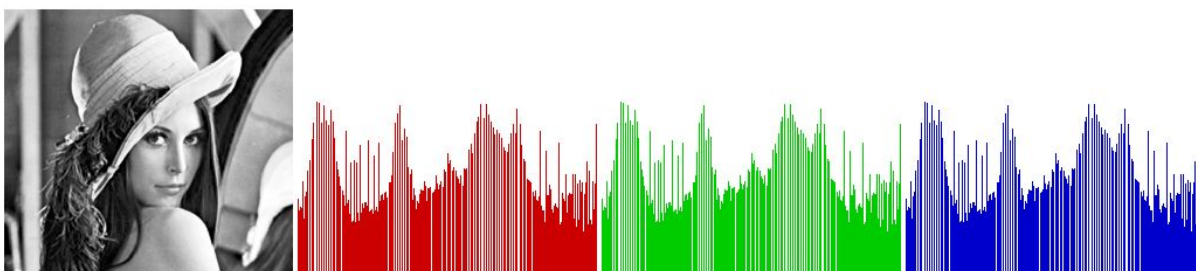
### 4. Metoda własna

Metodę oparliśmy na metodzie sąsiedztwa. W metodzie sąsiedztwa brana jest średnia wartość sąsiednich pikseli. Metoda własna bierze pod uwagę ich maksimum.

Oryginał



Modyfikacje



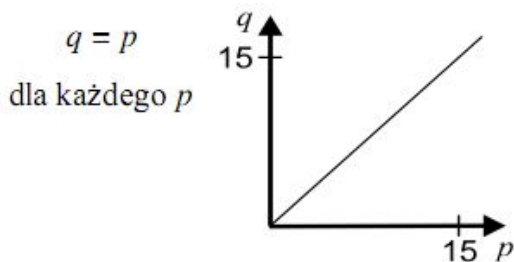
## Laboratorium 2: Operacje punktowe

### 1. Operacje punktowe jednoargumentowe

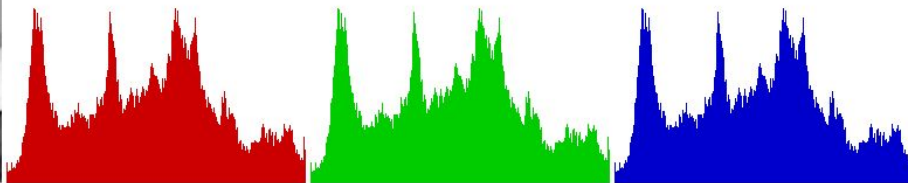
Wartość danego piksla  $a(i, j)$  obrazu pierwotnego ma wpływ na wartość tylko jednego piksla  $b(i, j)$  obrazu wynikowego.

#### Negacja

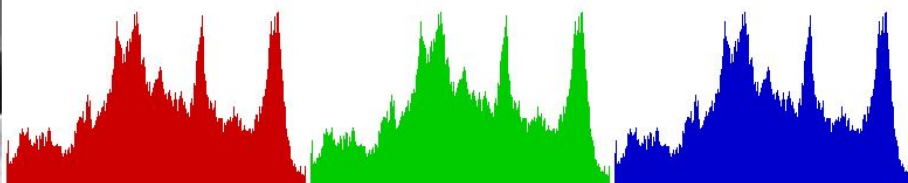
Histogram obrazu wynikowego stanie się lustrzanym odbiciem histogramu pierwotnego. Obraz wynikowy ma wygląd zbliżony do negatywu.



Oryginał



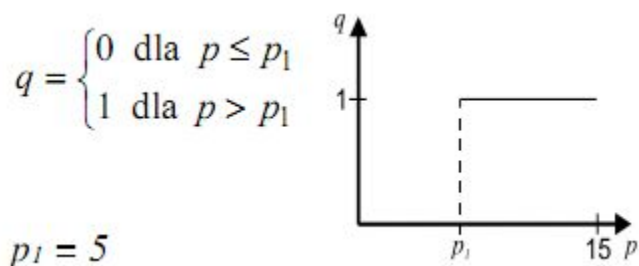
Modyfikacje



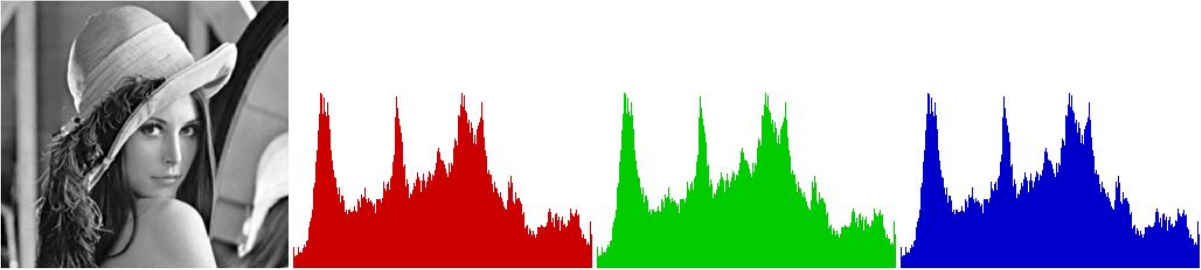
#### Progowanie

Każdy piksel obrazu pierwotnego jest przypisywany do jednej z dwóch kategorii:

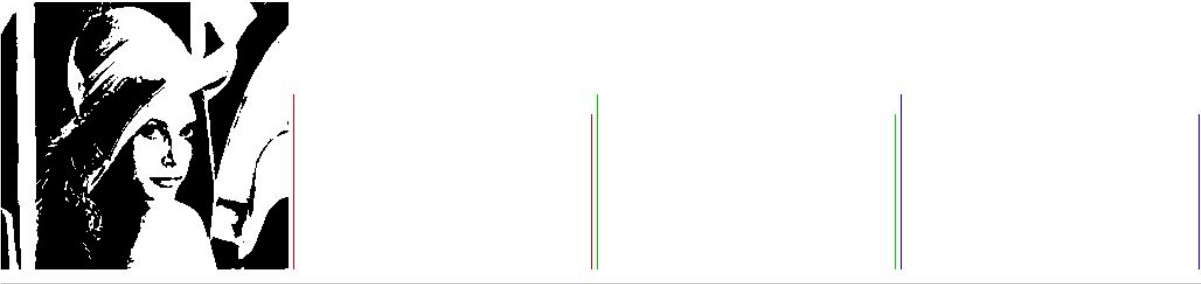
- 1) Pikseli czarnych: mniejszych lub równych wartości do wyznaczonego progu
- 2) Pikseli białych: większych wartości od wyznaczonego progu



Oryginał

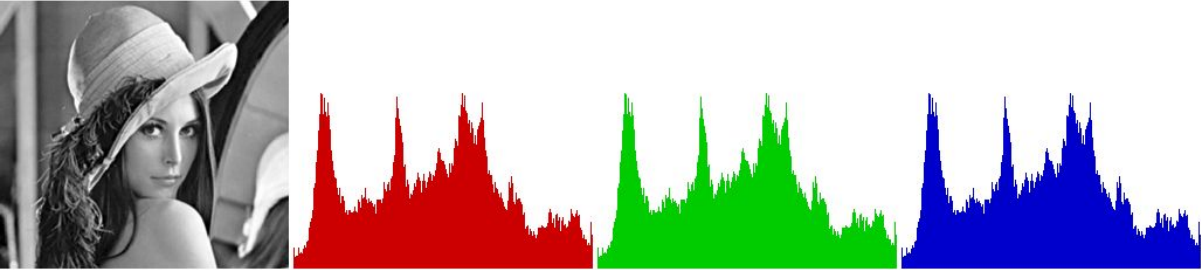


Modyfikacje

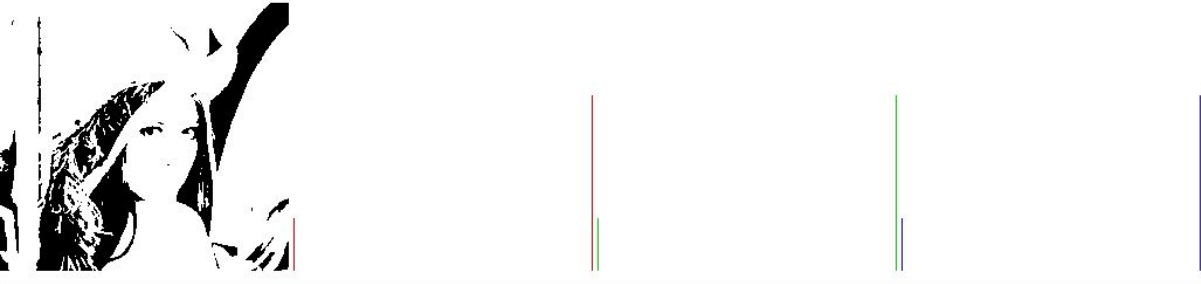


Poziom progu:  

Oryginał



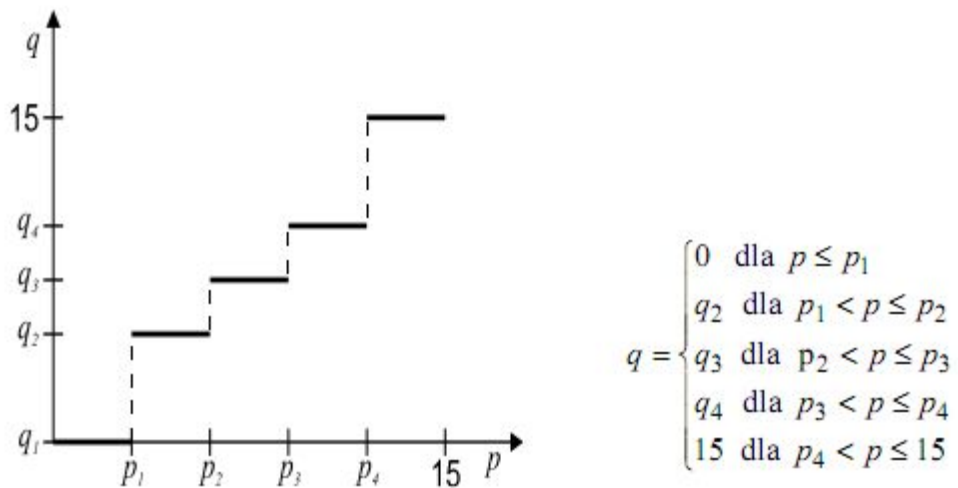
Modyfikacje



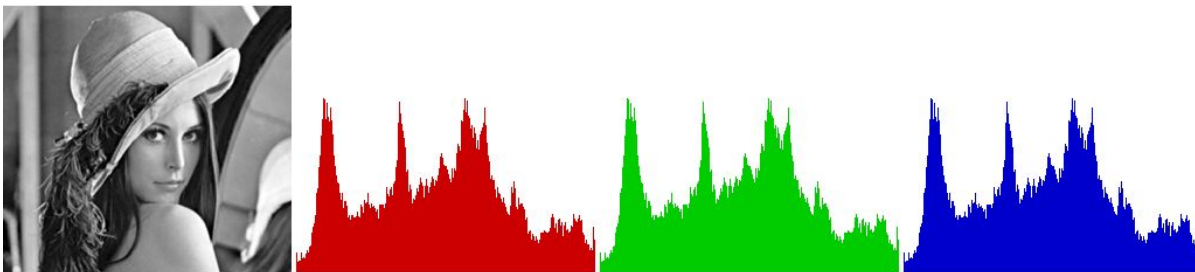
Poziom progu:  

## Redukcja poziomów szarości

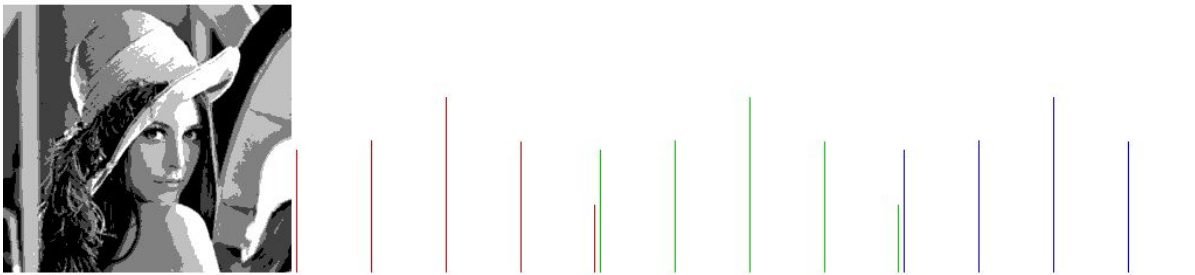
Operacja zbliżona do progowania, operuje jednak na większej liczbie kategorii pikseli. Dzięki temu zachowane są przejściowe poziomy między czernią i bielą.



Oryginał



Modyfikacje

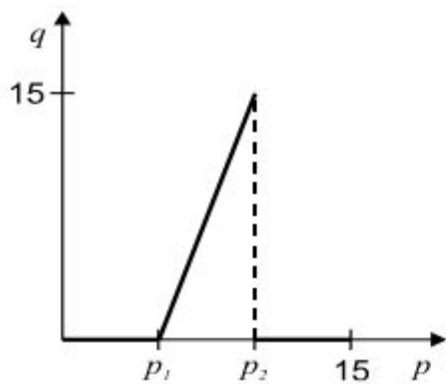


Poziomy:



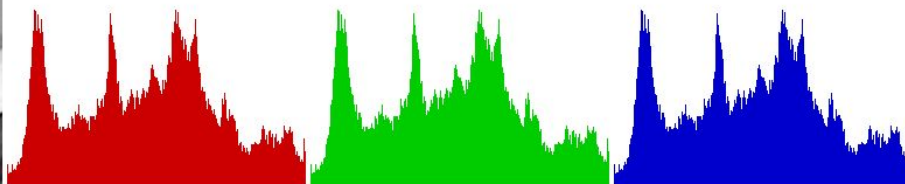
## Rozciąganie

Piksele, których wartości znajdują się w wyznaczonych przedziałach otrzymają nowe wartości. Obraz staje się bardziej kontrastowy.

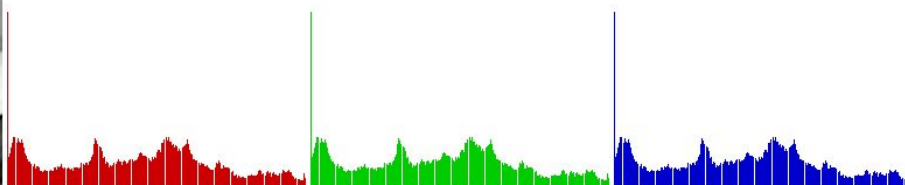


$$q = \begin{cases} (p - p_1) \frac{15}{(p_2 - p_1)} & \text{dla } p_1 < p \leq p_2 \\ 0 & \text{dla } p \leq p_1, p > p_2 \end{cases}$$

Oryginał

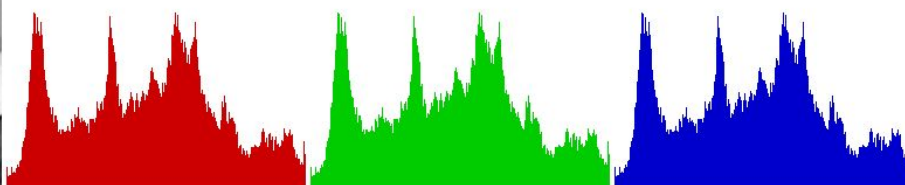


Modyfikacje



Początek:   Koniec:

Oryginał



Modyfikacje

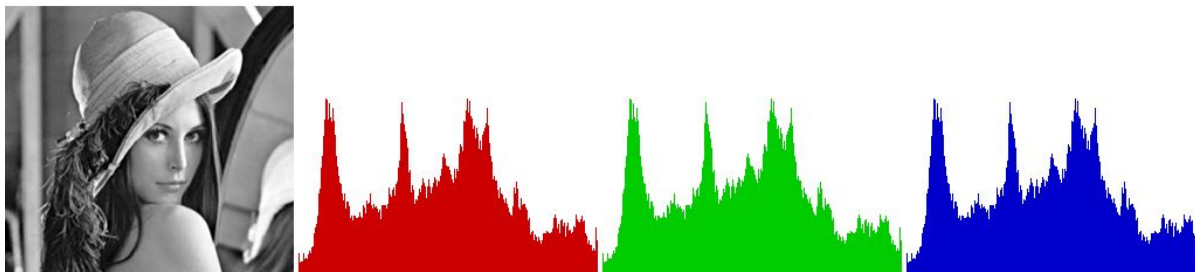


Początek:   Koniec:

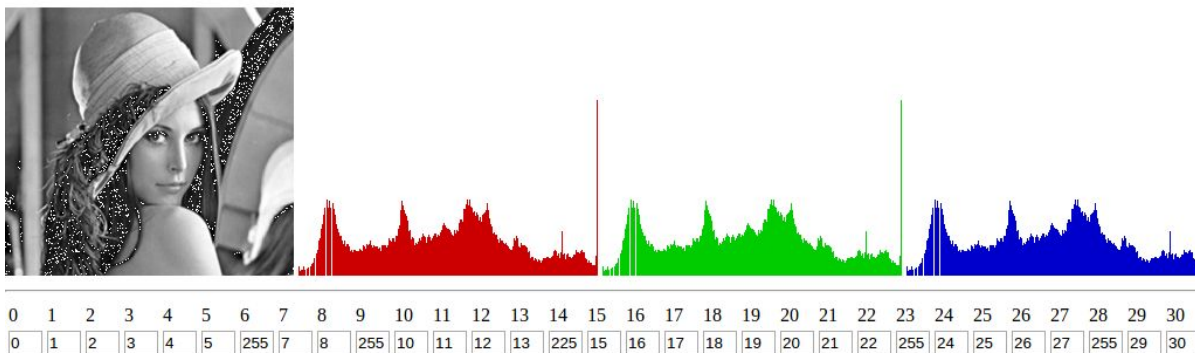
## 2. Uniwersalny Operator Punktowy

UOP pozwala przypisać dowolnemu poziomowi szarości inną wartość przy pomocy tablicy LUT. Przykład poniżej pokazuje przypisanie części poziomów wartości 255.

Oryginał



Modyfikacje

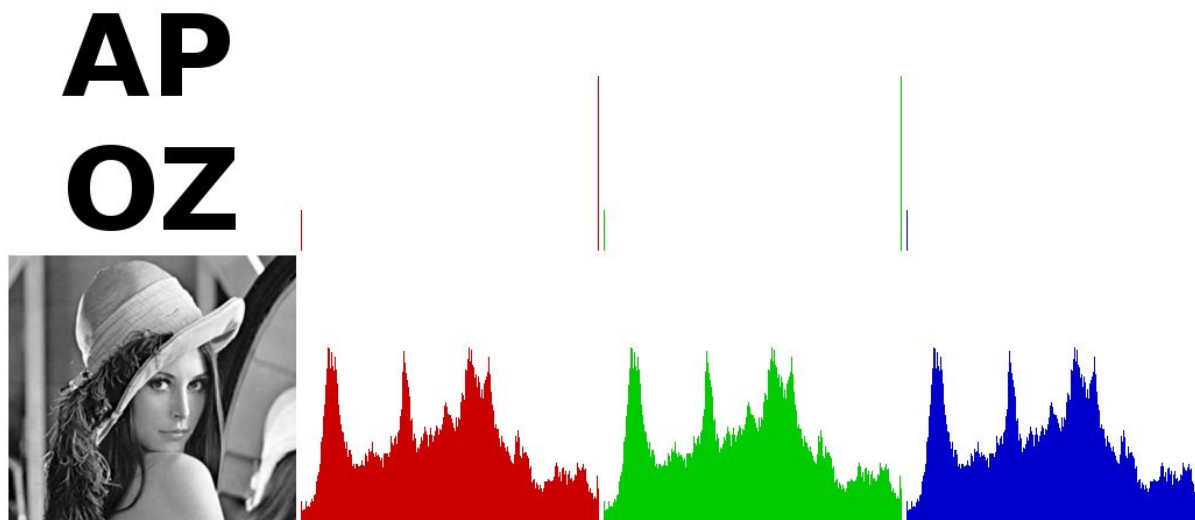


### 3. Operacje punktowe dwuargumentowe arytmetyczne i logiczne

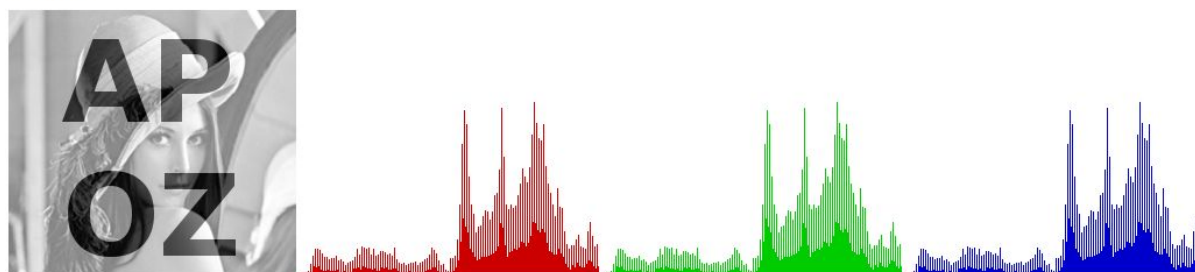
Obraz wynikowy jest rezultatem operacji na dwóch obrazach pierwotnych. Wartości poszczególnych pikseli pierwszego i drugiego obrazu mogą być: dodawane, odejmowane, mnożone. Możliwe jest również przeprowadzenie na nich operacji logicznych. Dokonują się one na bitowej reprezentacji wartości zestawionych pikseli.

#### Arytmetyczna suma

Oryginał

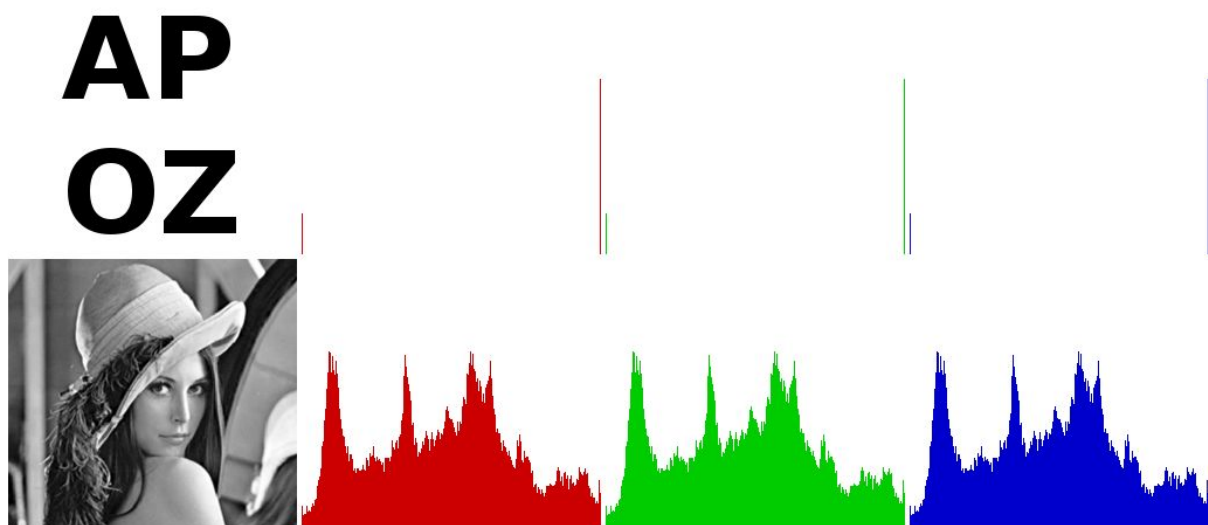


Modyfikacje

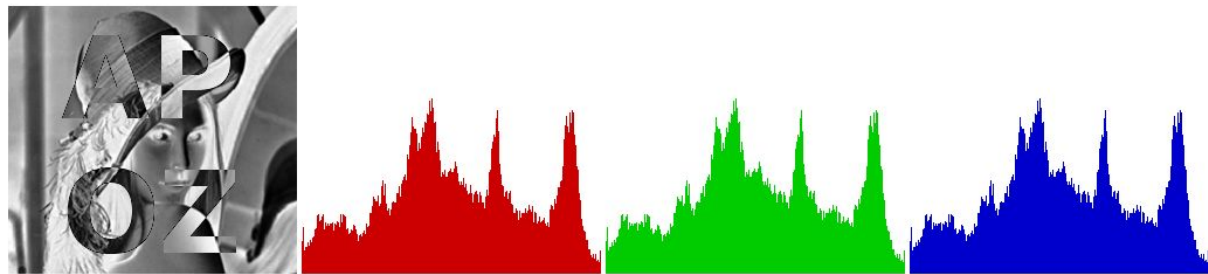


Arytmetyczna różnica

Oryginał

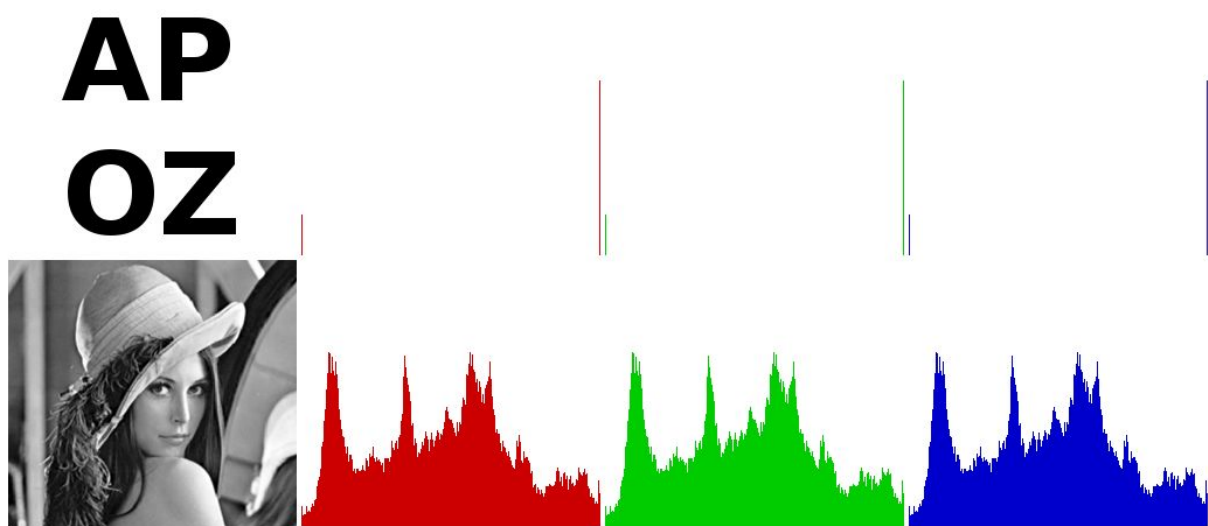


Modyfikacje

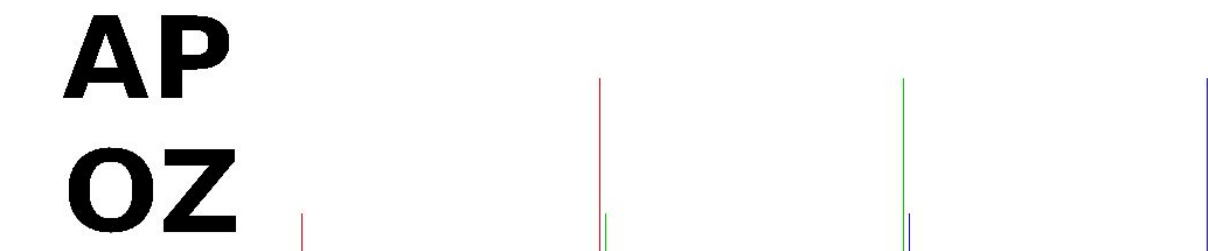


Arytmetyczny iloczyn

Oryginał

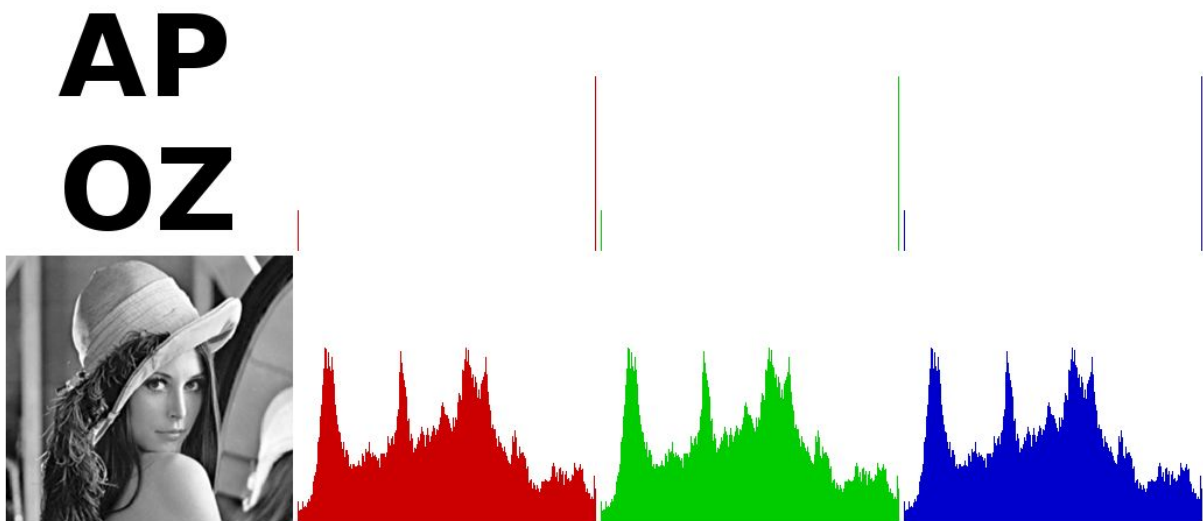


Modyfikacje

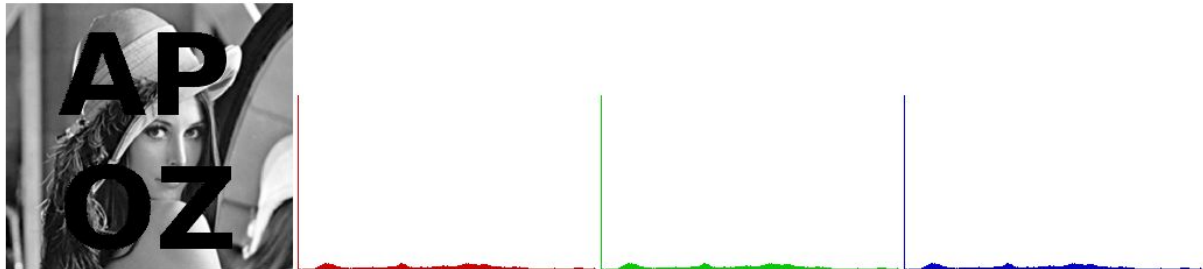


Logiczne AND

Oryginał

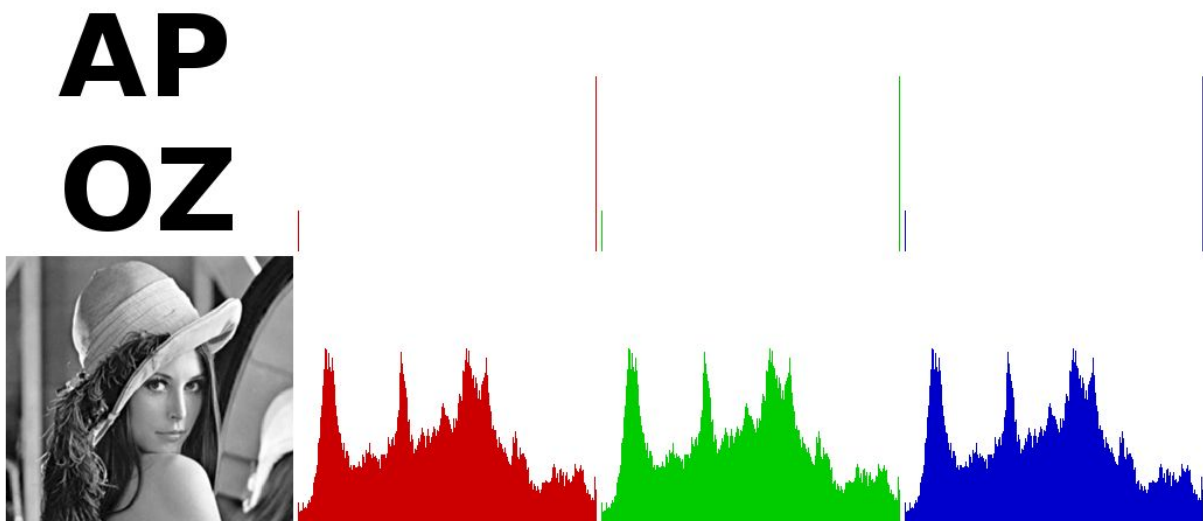


Modyfikacje

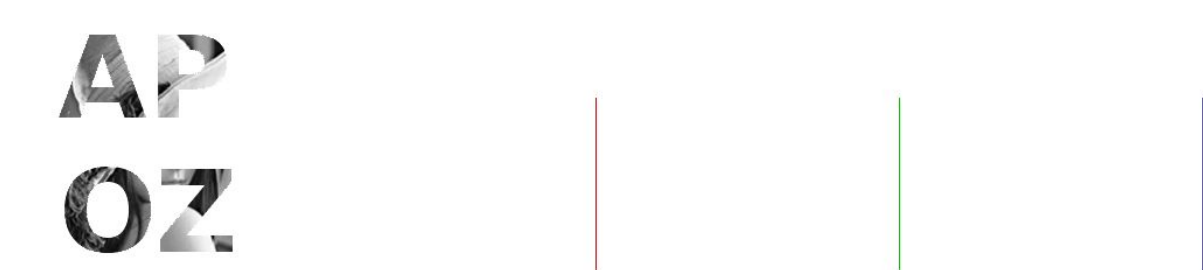


Logiczne OR

Oryginał

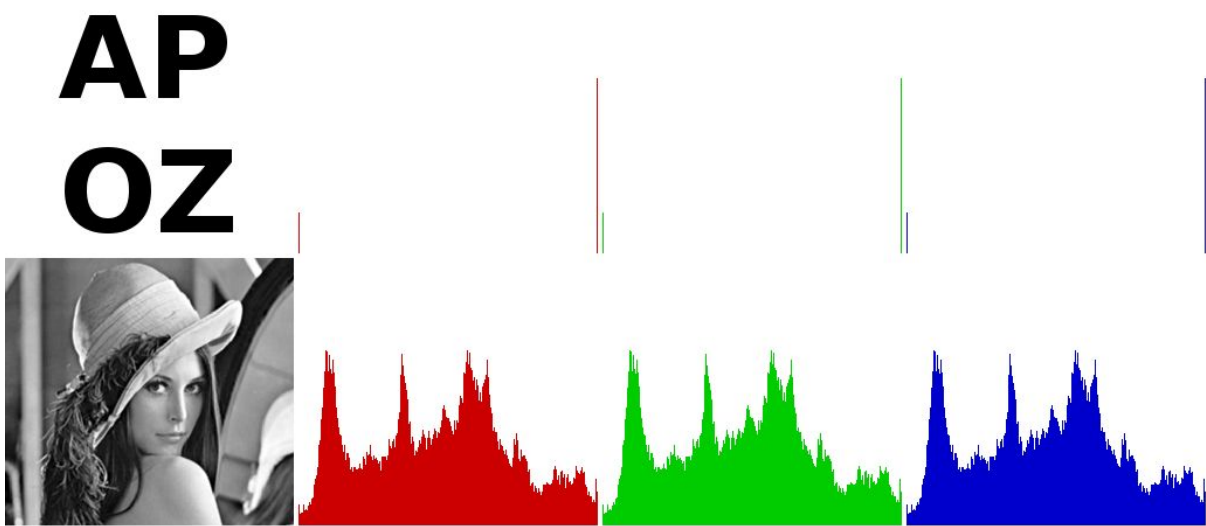


Modyfikacje

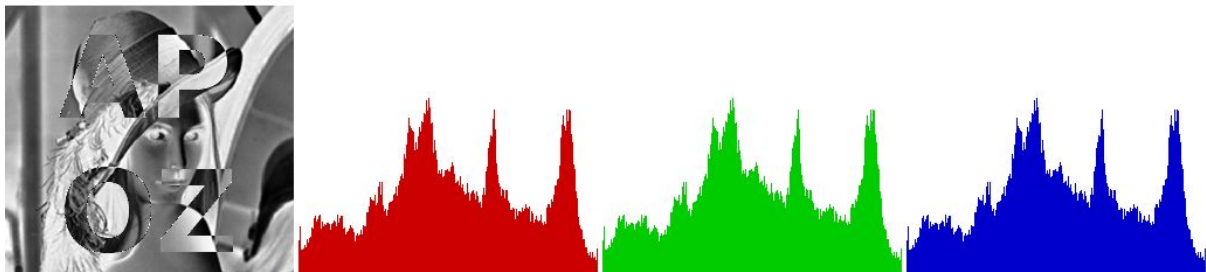


Logiczne XOR

Oryginał



Modyfikacje





# Laboratorium 3: Operacje sąsiedztwa

## 1. Filtracja liniowa

Aplikacja zawiera przygotowany zestaw masek do wygładzania (filtry dolnoprzepustowe), wyostrażania i detekcji krawędzi (filtry górnoprzepustowe). Możliwe jest stworzenie własnej maski.

Dostępne są opcje:

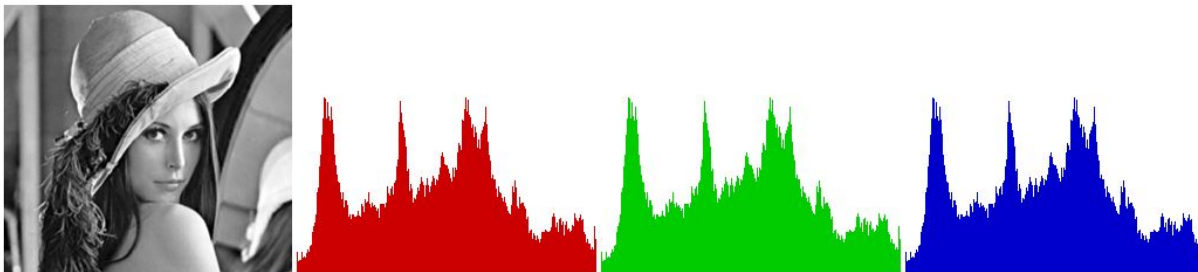
- Metody skalowania: proporcjonalna, trójwartościowa, obcinająca;
- Wyboru pikseli brzegowych: rozszerz, pomiń, istniejące.

### Wygładzenie

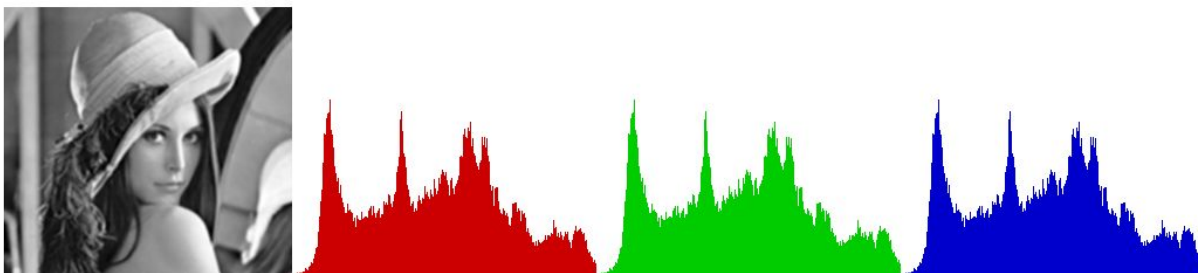
Zastosowano maskę silną o wartościach:

1	1	1
1	1	1
1	1	1

Oryginał



Modyfikacje



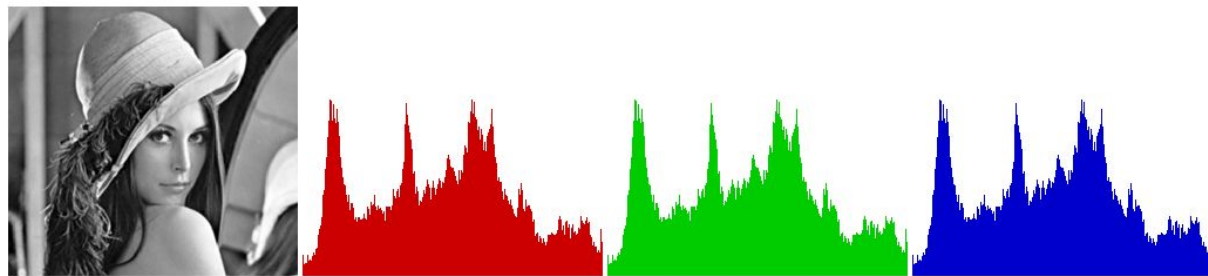
Wyostczenie

Zastosowano maskę silną o wartościach:

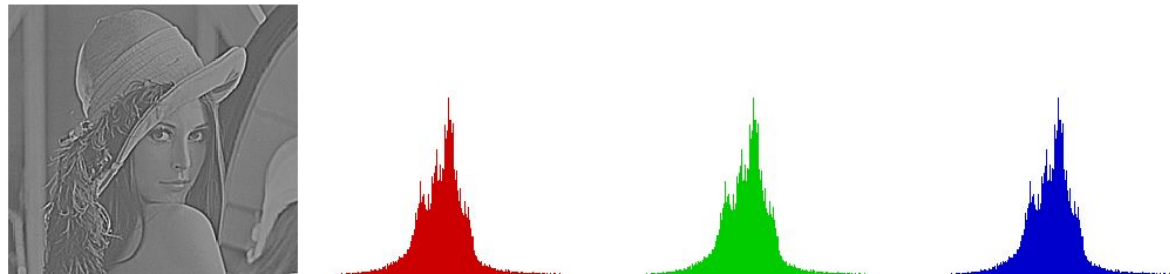
-1	-1	-1
-1	9	-1
-1	-1	-1

Metoda skalowania: Proporcjonalna

Oryginał

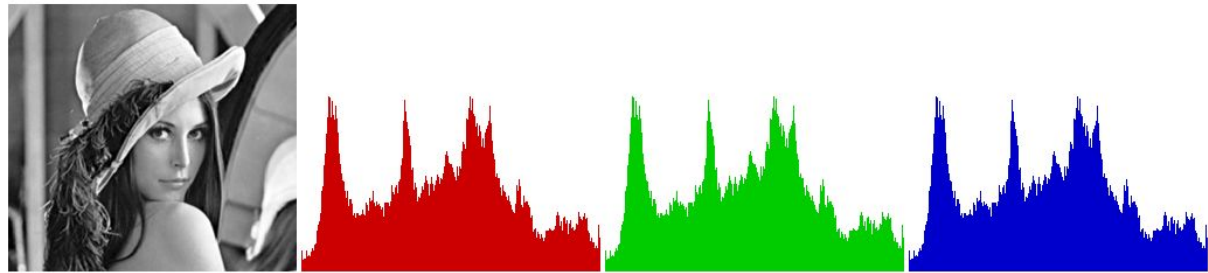


Modyfikacje

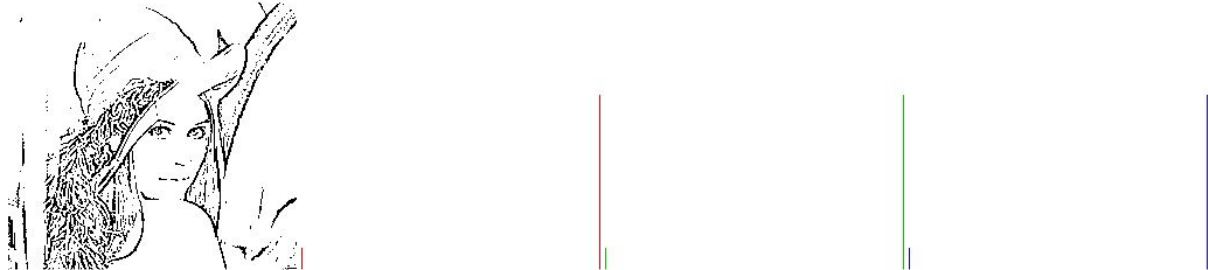


Metoda skalowania: Trójwartościowa

Oryginał



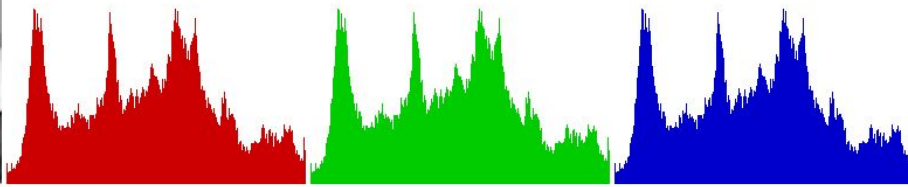
Modyfikacje



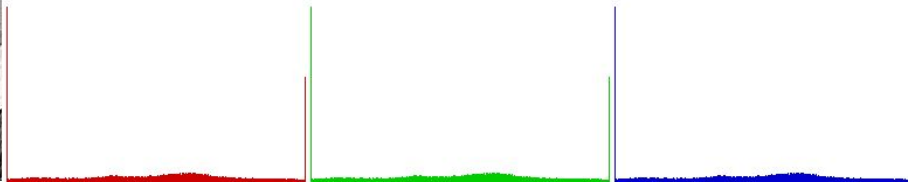


## Metoda skalowania: Obcinająca

Oryginał



Modyfikacje



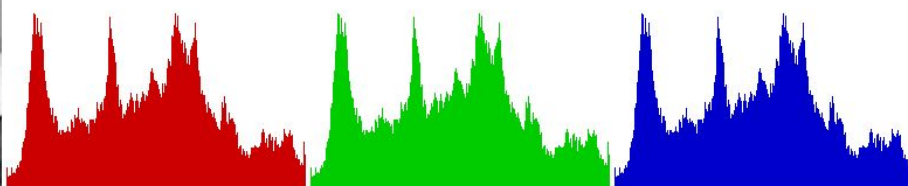
## Detekcja krawędzi

Zastosowano maskę silną o wartościach:

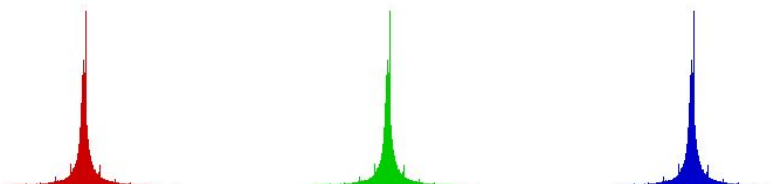
0	0	0
-1	1	0
0	0	0

## Metoda skalowania: Proporcjonalna

Oryginał

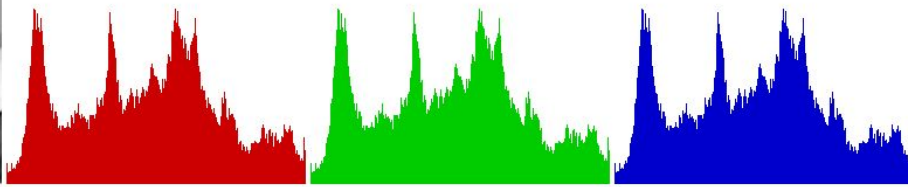


Modyfikacje



## Metoda skalowania: Trójwartościowa

Oryginał

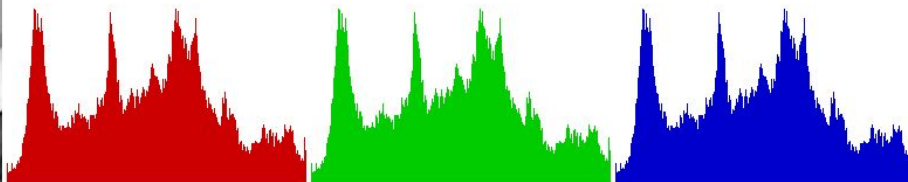


Modyfikacje

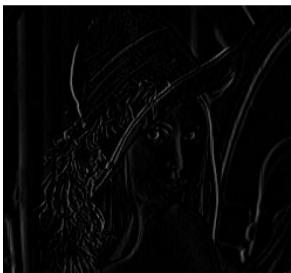


## Metoda skalowania: Obcinająca

Oryginał



Modyfikacje

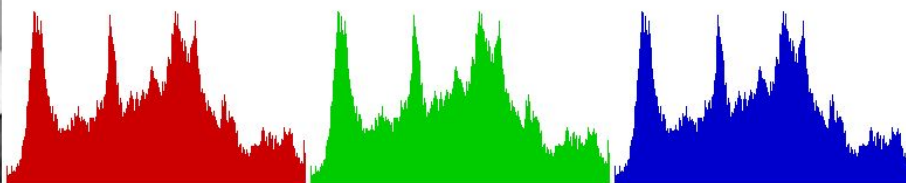


## 2. Filtry statyczne

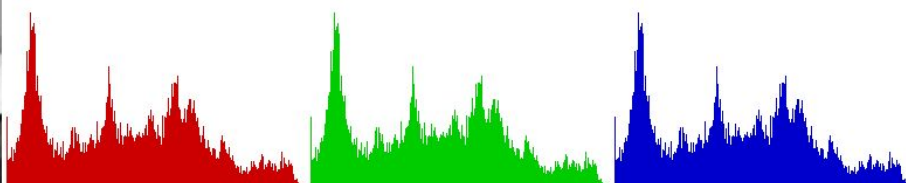
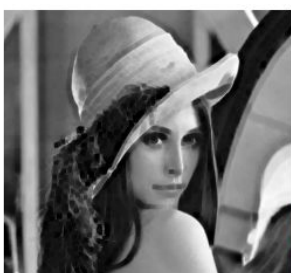
Filtry statyczne nie posiadają stałej maski. Wartości dla nowych pikseli wylicza się podczas sortowania pikseli obrazu źródłowego w obrębie maski. Ostateczna wartość nowego piksela wybierana jest po sortowaniu zgodnie w użytym filtrem. Może to być wartość środkowa, maksymalna, minimalna.

### Minimalny

Oryginał

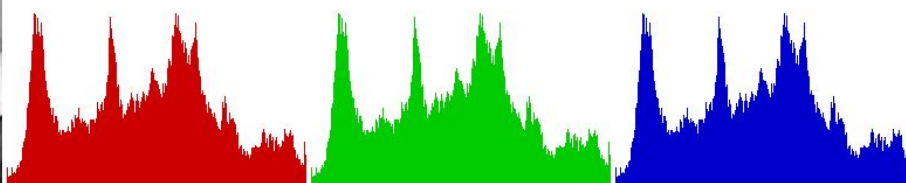


Modyfikacje

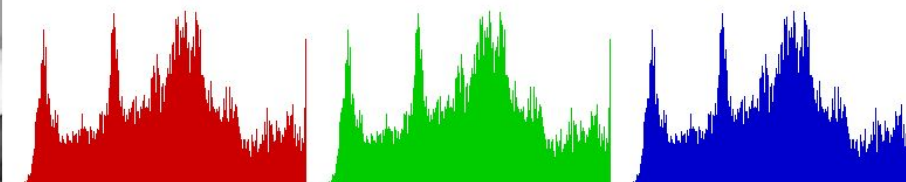


### Maksymalny

Oryginał

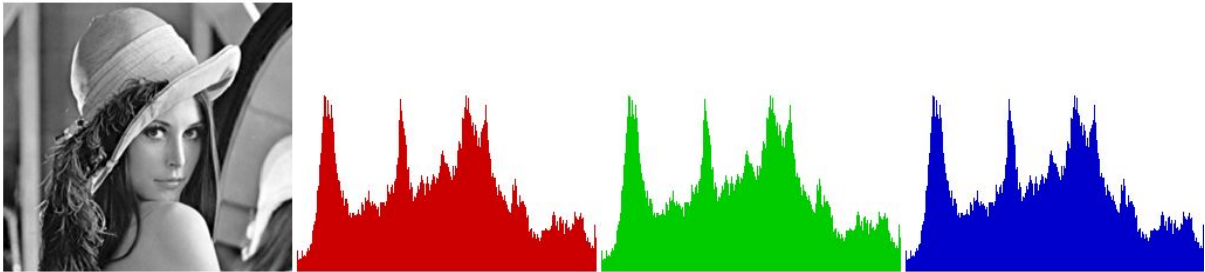


Modyfikacje

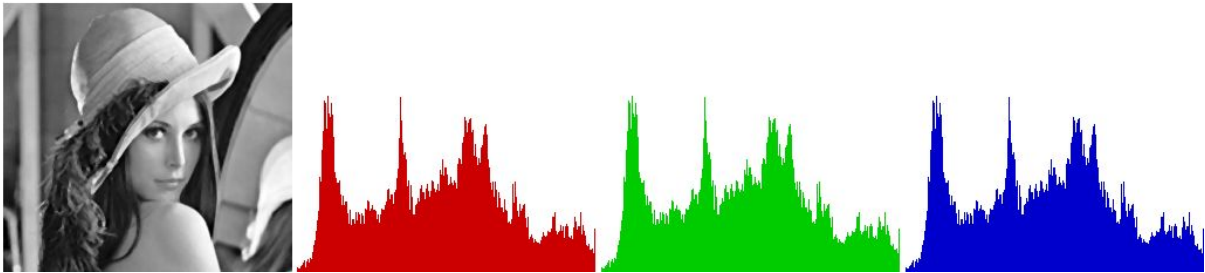


Medianowy

Oryginał



Modyfikacje



# Laboratorium 4: Operacje sąsiedztwa

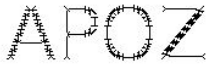
## 1. Ścienianie

Pozwala odtworzyć liniową strukturę obrazu bez niszczenia jego spójności.

Oryginał



Modyfikacje



## 2. Operacje morfologiczne erozji, dylatacji, otwarcia, zamknięcia

Zmieniają strukturę obiektów na obrazie. Najlepiej sprawdzają się przy obrazach binarnych (mających dwa kolory).

### Erozja

Polega na obcinaniu brzegów obiektu na obrazie. Wada: zmniejszanie obiektów.

Oryginał



Modyfikacje



## Dylatacja

Zamyka małe otwory i zatoki figury. Dzięki temu obiekty na obrazie zwiększają swoją objętość. Zrastają się jeśli są blisko siebie. Wada: zwiększanie się obiektów.

Oryginał



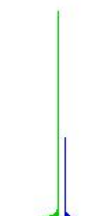
Modyfikacje



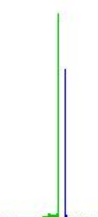
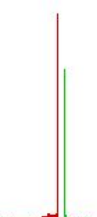
## Otwarcie

Eliminuje wady erozji zestawiając ją z dylatacją.

Oryginał



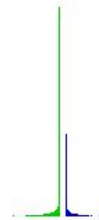
Modyfikacje



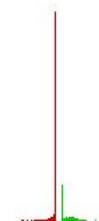
## Zamknięcie

Eliminuje wady dylatacji zestawiając ją z erozją.

Oryginał



Modyfikacje



## Laboratorium 5: algorytmy segmentacji i analizy obrazu

### 1. Algorytm żółwia

Oryginał

**AP**  
**OZ**

Modyfikacje

**A****P**  
**O****Z**



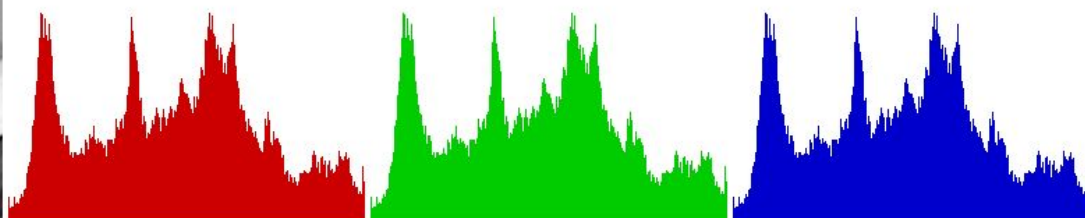
# Projekt: Konwersja obrazów szarych na kolorowe

## 1. Omówienie

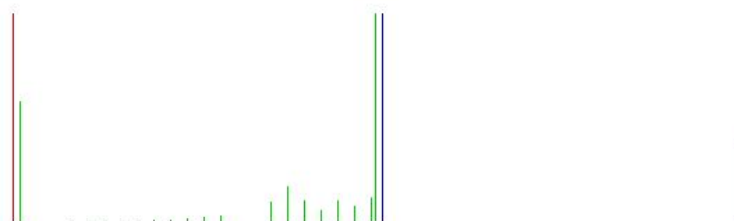
Aplikacja pozwala przeprowadzić operację koloryzacji obrazów szarych na kolorowe. Transformacja odbywa się na podstawie danych dostarczonych przez funkcję zmiany koloru. Jej wartości można zmienić za pomocą wykresu o 128 wartościach na osi x i y. Mapuje on stare wartości (oś x) na nowe (oś y). Obrazy wynikowe prezentowane są po naciśnięciu przycisku "Koloryzuj".

Najechnanie kursorem myszy na obszar obrazu wynikowego spowoduje wyświetlenie wartości HSV wskazanego punktu. Informacja ta znajduje się na kolorowym tle odzwierciedlającym barwę wybranego piksela.

Oryginał

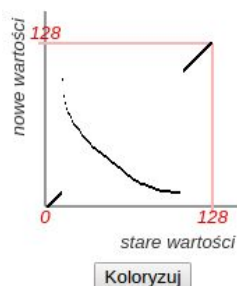


Modyfikacje

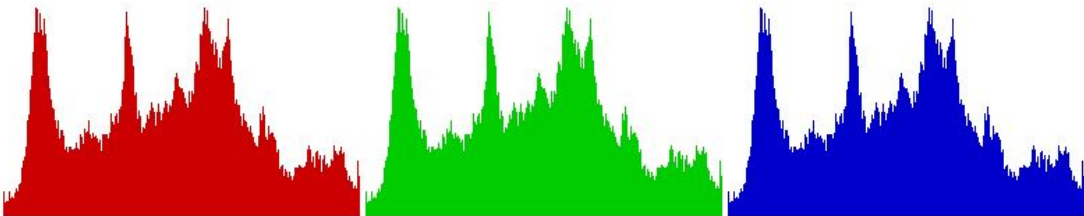


Narysuj funkcję zmiany koloru na wykresie: zacznij od lewej strony trzymając przyciśnięty klawisz myszy

Wartości wskazanego punktu: **H: 0.16470588235294117, S: 1, V: 1, x: 139, y: 206**



Oryginał

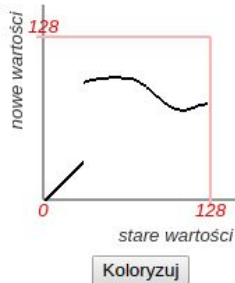


Modyfikacje

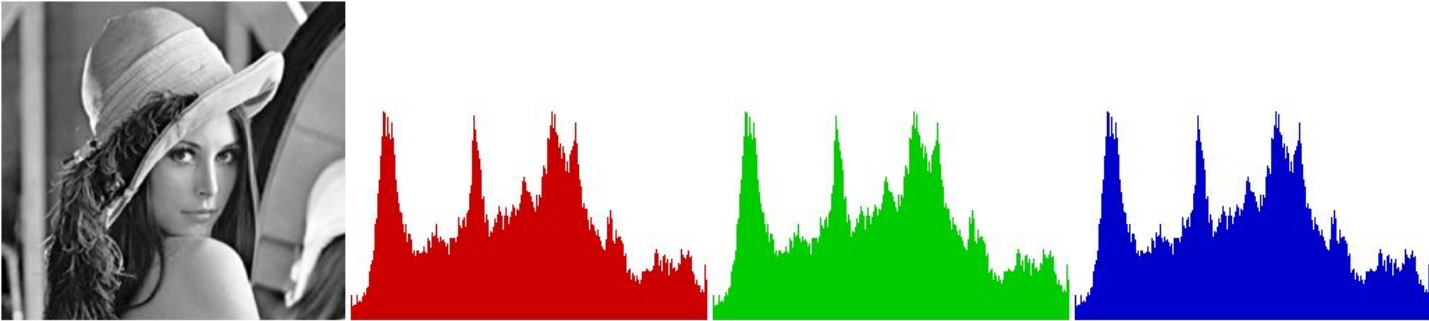


Narysuj funkcję zmiany koloru na wykresie:  
zaczynij od lewej strony trzymając przyciśnięty  
klawisz myszy

Wartości wskazanego punktu: **H: 0.8333333333333334, S: 1, V: 1, x: 46, y: 27**



Oryginał

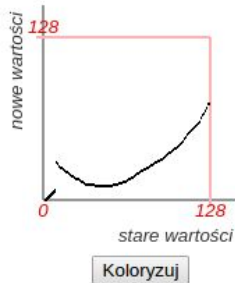


Modyfikacje



Narysuj funkcję zmiany koloru na wykresie:  
zaczynij od lewej strony trzymając przyciśnięty  
klawisz myszy

Wartości wskazanego punktu: **H: 0.333333333333333, S: 1, V: 1, x: 141, y: 102**



## 2. Algorytm koloryzacji

Kluczowym miejscem algorytmu jest pętla for. Generowane są tu nowe wartości dla pikseli obrazu w krokach:

1. Wartości dostarczone przez funkcję zmiany koloru są w zakresie 1-128. Oryginalny obraz w odcieniach szarości ma wartości w zakresie 0-255 dla RGB. Następuje przełożenie wartości piksela starego obrazu na wartość funkcji zmiany.
2. Nowy kolor będzie wygenerowany zależnie od dostarczonej nowej wartości piksela. Zakres wartości został podzielony na sekcje, każda wyznacza trochę inny sposób generowania koloru. Dzięki temu w wielu wypadkach zachowamy charakterystyczne obiekty na obrazie, a ich nowe kolory będą wyraziste. Sekcje oddalone są od siebie o 22 wartości piksela.
3. Dodanie wyznaczonych wartości do kanałów obrazu kolorowego Red, Green, Blue.

```
for (let i = 0; i < pixels.length; i++) {
  let previousColor = pixels.data[i];

  // switching from 256 to 128
  let newColorIndex = Math.floor(previousColor / 2) + previousColor % 2;
  let newColorObj = params[newColorIndex];
  let newColor = newColorObj.Y;

  if (newColor < 22) {
    newPixelsR.data[i] = 255;
    newPixelsG.data[i] = 12 + 12 * newColor;
    newPixelsB.data[i] = 0;
  } else if (newColor < 44) {
    newPixelsR.data[i] = 255 - (12 + 12 * newColor);
    newPixelsG.data[i] = 255;
    newPixelsB.data[i] = 0;
  } else if (newColor < 66) {
    newPixelsR.data[i] = 0;
    newPixelsG.data[i] = 255;
    newPixelsB.data[i] = 12 + 12 * newColor;
  } else if (newColor < 88) {
    newPixelsR.data[i] = 0;
    newPixelsG.data[i] = 225 - (12 + 12 * newColor);
    newPixelsB.data[i] = 255;
  } else if (newColor < 110) {
    newPixelsR.data[i] = 12 + 12 * newColor;
    newPixelsG.data[i] = 0;
    newPixelsB.data[i] = 255;
  } else {
    newPixelsR.data[i] = 255;
    newPixelsG.data[i] = 0;
    newPixelsB.data[i] = 255 - (12 + 12 * newColor);
  }
}

transformedPicture.putPixels(newPixelsR, 'red');
transformedPicture.putPixels(newPixelsG, 'green');
transformedPicture.putPixels(newPixelsB, 'blue');
};
```

## Literatura

- Materiały udostępnione w ramach przedmiotów APO i POB
- M. Doros, Przetwarzanie obrazów, Skrypt WSISIZ
- I.Pitas, Digital Image processing Algorithms And Applications
- W. Skarbek, Metody reprezentacji obrazów cyfrowych
- R.Tadeusiewicz, P.Korohoda, Komputerowa analiza i przetwarzanie obrazów, Kraków 1997
- [https://pl.wikipedia.org/wiki/Filtracja\\_obrazów](https://pl.wikipedia.org/wiki/Filtracja_obrazów) (dostęp 21-01-2017)
- <http://atol.am.gdynia.pl/tc/Radzienski/morfologiczne.htm> (dostęp 21-01-2017)