

Uniwersytet Rzeszowski
Wydział Nauk Ścisłych i Technicznych
Instytut Informatyki



BIOMETRYCZNE SYSTEMY ZABEZPIECZEŃ

INSTRUKCJA DO ĆWICZEŃ LABORATORYJNYCH

TREŚCI KSZTAŁCENIA: WPROWADZENIE DO BIBLIOTEKI MATLAB IMAGE PROCESSING ORAZ HISTOGRAMOWE METODY SEGMENTACJI OBRAZÓW.

Spis treści

1. Cele laboratorium	2
2. Biblioteka Image Processing Toolbox	2
3. Histogram i metody segmentacji obrazów	6
4. Zadania do samodzielnego rozwiązania	12

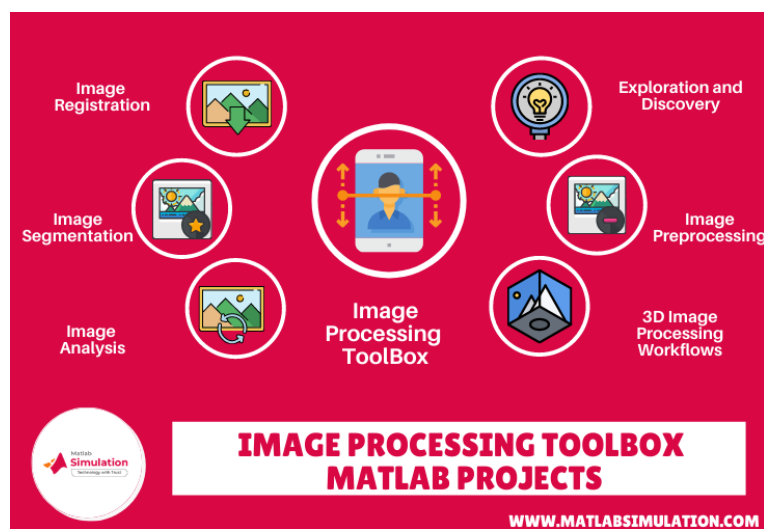
1. Cele laboratorium

Celem laboratorium jest zapoznanie się z podstawowymi funkcjami biblioteki Image Processing Toolbox w MATLAB-ie oraz zastosowanie metod segmentacji obrazów opartych na analizie histogramu. Należy zapoznać się z materiałami i przykładami zawartymi w niniejszych materiałach oraz zaproponować rozwiązania dla zadań do samodzielnego rozwiązania. Rozwiązania należy umieścić w sprawozdaniu, zgodnie z wymaganiami.

2. Biblioteka Image Processing Toolbox

Wprowadzenie

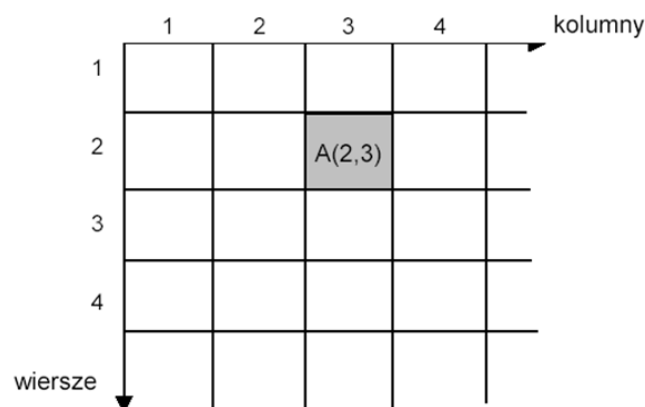
Image Processing Toolbox (IPT) to zbiór funkcji w MATLAB-ie przeznaczonych do analizy i przetwarzania obrazów. Obejmuje on funkcje do operacji na obrazach, takie jak filtrowanie, transformacje geometryczne, segmentacja oraz analiza cech. Szczegółowe informacje o bibliotece wraz z przykładami dostępne są pod adresem <https://uk.mathworks.com/products/image-processing.html>



Rys. 1. Schemat dostępnych funkcji w Image Processing Toolbox

Struktury danych stosowane do reprezentacji i przetwarzania obrazów w bibliotece Image Processing Toolbox

Podstawowe typy obrazów monochromatycznych są reprezentowane w pakiecie MATLAB za pomocą tablic dwuwymiarowych. Każdy element tablicy odpowiada jednemu punktowi obrazu cyfrowego określanego jako piksel.



Rys. 2 Obraz jako tablica dwuwymiarowa

W pakiecie MATLAB, elementy tablic są standardowo reprezentowane za pomocą 64 bito-wych liczb zmiennoprzecinkowych (klasa double). W bibliotece IPT stosuje się głównie klasę uint8, tj. zmienną ośmiobitową bez znaku.

Zmienne klasy uint8

Biblioteka IPT udostępnia ograniczony zbiór działań na zmiennych klasy uint8, tj.:

- wyświetlanie obrazów reprezentowanych za pomocą zmiennej uint8

Przykład:

```
% Wczytanie obrazu w formacie uint8
I = imread('cameraman.tif');

% Sprawdzenie klasy danych
disp(class(I)); % Powinno zwrócić 'uint8'

% Wyświetlenie obrazu
figure, imshow(I), title('Obraz w formacie uint8');
```

- indeksowanie (adresowanie punktów) obrazów,

```
% Wczytanie obrazu
I = imread('cameraman.tif');

% Pobranie wartości piksela w wierszu 50 i kolumnie 100
pixel_value = I(50,100);
disp(['Wartość piksela (50,100): ', num2str(pixel_value)]);

% Zmiana wartości piksela na maksymalną (255)
I(50,100) = 255;

% Wyświetlenie zmodyfikowanego obrazu
figure, imshow(I), title('Obraz po modyfikacji piksela');
```

- zmianę wymiarów tablic i kolejności elementów tablic, m.in. za pomocą poleceń: reshape, cat, permute.
a) Zmiana wymiarów tablicy za pomocą reshape

```
% Zmiana wartości piksela na maksymalną (255)
I(50,100) = 255;

% Wyświetlenie zmodyfikowanego obrazu
figure, imshow(I), title('Obraz po modyfikacji piksela');
```



```
% Wczytanie obrazu
I = imread('cameraman.tif');
```

```
% Przekształcenie obrazu na wektor kolumnowy
I_resaped = reshape(I, [], 1);

% Wyświetlenie rozmiaru przed i po zmianie
disp(['Rozmiar oryginalny: ', num2str(size(I))]);
disp(['Rozmiar po reshape: ', num2str(size(I_resaped))]);
```

b) Łączenie tablic (cat)

```
% Wczytanie obrazu
I = imread('cameraman.tif');

% Połączenie obrazu z jego kopią w poziomie (concat wzdłuż drugiego wymiaru)
I_cat = cat(2, I, I);

% Wyświetlenie połączanego obrazu
figure, imshow(I_cat), title('Obraz po połączeniu w poziomie');
```

c) Zamiana kolejności wymiarów (permute)

```
% Wczytanie obrazu RGB
RGB = imread('peppers.png');

% Zamiana kolejności wymiarów (np. zamiana osi 1 i 2)
RGB_permuted = permute(RGB, [2, 1, 3]);

% Wyświetlenie oryginalnego i permutowanego obrazu
figure, subplot(1,2,1), imshow(RGB), title('Oryginalny obraz RGB');
subplot(1,2,2), imshow(RGB_permuted), title('Obraz po permutacji wymiarów');
```

Niezależnie od klasy zmiennej stosowanej do kodowania jasności punktu obrazu, w bibliotece IPT stosuje się cztery podstawowe struktury danych do reprezentacji obrazów:

- obrazy indeksowane {indexed images} są zapisywane jako para: tablica indeksów (dwuwymiarowa macierz X) oraz mapa kolorów (map), będąca tablicą N×3 określającą wartości kolorów RGB.

Przykład. Konwersja obrazu RGB na obraz indeksowany

```
% Wczytanie obrazu RGB
RGB = imread('peppers.png');
% Konwersja obrazu RGB na obraz indeksowany (z 256 kolorami)
[X, map] = rgb2ind(RGB, 256);
% Wyświetlenie obrazu indeksowanego
figure, imshow(X, map), title('Obraz indeksowany');
% Sprawdzenie struktury danych
disp(['Rozmiar indeksów: ', num2str(size(X))]);
disp(['Rozmiar mapy kolorów: ', num2str(size(map))]);
```

- obrazy monochromatyczne (intensity images, intensywności) są zapisane jako dwuwymiarowa macierz wartości jasności. W MATLAB-ie są zwykle w formacie uint8 lub double.

Przykład. Wczytanie i konwersja do obrazu monochromatycznego.

```
% Wczytanie obrazu RGB
RGB = imread('peppers.png');
% Konwersja do obrazu w skali szarości
I = rgb2gray(RGB);
% Wyświetlenie obrazu
figure, imshow(I), title('Obraz monochromatyczny');
% Sprawdzenie zakresu wartości pikseli
disp(['Minimalna wartość piksela: ', num2str(min(I(:)))]);
disp(['Maksymalna wartość piksela: ', num2str(max(I(:)))]);
```

- obrazy binarne (binary images), zawierają tylko dwa poziomy jasności: 0 (czarny) i 1 (biały). Są często używane w segmentacji i detekcji obiektów.

Przykład. Konwersja obrazu do formatu binarnego

```
% Wczytanie obrazu w skali szarości
I = imread('cameraman.tif');
% Binarizacja metodą Otsu
level = graythresh(I);
BW = imbinarize(I, level);
% Wyświetlenie obrazu binarnego
figure, imshow(BW), title('Obraz binarny');
% Sprawdzenie unikalnych wartości w obrazie binarnym
disp(['Unikalne wartości pikseli: ', num2str(unique(BW(:)))]);
```

- obrazy RGB (RGB images), są zapisane jako tablica trójwymiarowa $M \times N \times 3$, gdzie każda warstwa odpowiada wartościom dla jednego z kanałów: czerwonego, zielonego i niebieskiego.

Przykład: Wczytanie obrazu RGB i analiza kanałów

```
% Wczytanie obrazu RGB
RGB = imread('peppers.png');
% Wyodrębnienie poszczególnych kanałów kolorów
R = RGB(:, :, 1); % Kanał czerwony
G = RGB(:, :, 2); % Kanał zielony
B = RGB(:, :, 3); % Kanał niebieski
% Wyświetlenie oryginalnego obrazu oraz poszczególnych kanałów
figure,
subplot(2,2,1), imshow(RGB), title('Oryginalny obraz RGB');
subplot(2,2,2), imshow(R), title('Kanał czerwony');
subplot(2,2,3), imshow(G), title('Kanał zielony');
subplot(2,2,4), imshow(B), title('Kanał niebieski');
% Sprawdzenie wymiarów obrazu RGB
disp(['Rozmiar obrazu RGB: ', num2str(size(RGB))]);
```

Obrazy w różnej głębi barw

Obrazy monochromatyczne (intensity images)

Obrazy monochromatyczne są reprezentowane w bibliotece IPT za pomocą pojedynczej tablicy, której elementy odpowiadają jasności poszczególnych punktów obrazu. Tablica obrazu monochromatyczny

może być klasy double lub uint8. Dla klasy double elementy tablicy przyjmują wartości z zakresu [0, 1], zaś dla tablicy klasy uint8 wartości całkowite z zakresu [0, 255].

Obrazy binarne (binary images)

Punkty obrazu binarnego przyjmują jedną z dwóch dyskretnych wartości. Podobnie jak dla obrazów monochromatycznych tablice reprezentujące obrazy binarne są klasy double lub uint8. Zaleca się stosowanie głównie klasy uint8 ze względu na oszczędność pamięci. Funkcje biblioteki IPT, które w wyniku zwracają obrazy binarne stosują klasę uint8.

Obrazy RGB (RGB images)

Obraz RGB jest definiowany przez trzy oddzielne tablice, każda o wymiarach odpowiadających wymiarowi obrazu. Tablice te zawierają intensywności kolorów składowych kolejno: czerwonego (Red), zielonego (Green) i niebieskiego (Blue). Zatem obraz RGB jest tablicą trójwymiarową $M \times N \times 3$, gdzie M jest liczbą wierszy a N liczbą kolumn obrazu, a trzeci wymiar tablicy wskazuje na składową koloru. Zatem kolor każdego punktu obrazu jest wynikiem złożenia trzech kolorów składowych. Tablica obrazu RGB jest klasy double lub uint8. Dla tablicy klasy double elementy tablicy przyjmują wartości z zakresu [0, 1], zaś dla tablicy klasy uint8 wartości całkowite z zakresu [0, 255].

3. Histogram i metody segmentacji obrazów

Histogram to wykres przedstawiający rozkład wartości pikseli w obrazie. Jest to narzędzie statystyczne, które pozwala na graficzną interpretację luminancji (jasności) obrazu. Histogramy mogą być stosowane do:

- Poprawy kontrastu – np. poprzez rozciąganie histogramu (histogram stretching) lub wyrównywanie histogramu (histogram equalization).
- Segmentacji obrazów – wybór progów dla segmentacji binarnej (np. metoda Otsu).
- Analizy jakości obrazu – identyfikacja prześwietlonych lub niedoświetlonych obszarów.
- Przetwarzania kolorów – analiza składowych RGB dla obrazów barwnych.

Histogram wyrażamy jako dyskretną funkcję określoną dla naturalnych wartości zmiennej k , której wartościami są ilości punktów o danym poziomie szarości (kolorze) k .

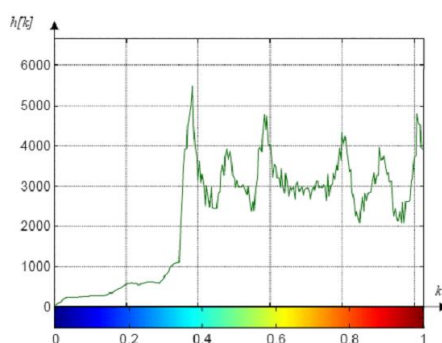
$$h = \{h[0], h[1], \dots, h[K - 1], \}$$
 (1)

gdzie K – liczba poziomów szarości w obrazie, zaś $h[k]$ określone są jako:

$$h[k] = \sum_{x=1}^M \sum_{y=1}^N w_{xy}$$
 (2)

$$w_{xy} = \begin{cases} 1 & \text{dla } I(x, y) = k \\ 0 & \text{inaczej} \end{cases}$$
 (3)

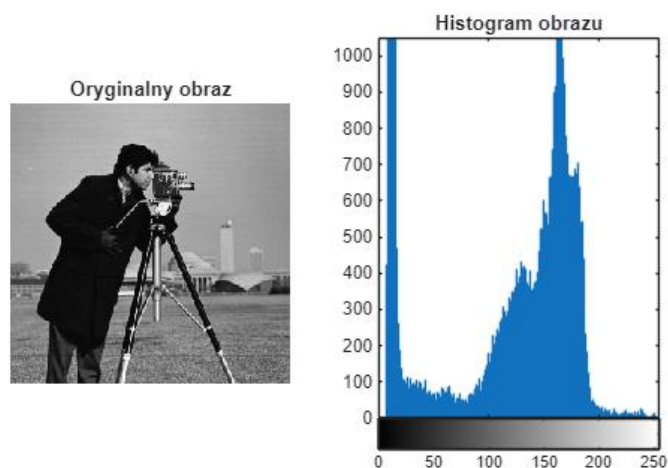
gdzie: M, N – rozmiar obrazu, $I(x, y)$ poziom intensywności obrazu w punkcie (x, y) .



Rys. 3 Przykładowa analiza

Każdy obraz cyfrowy pozwala na wykonanie prostej analizy statystycznej -histogramu, czyli wykresu który pokazuje, jak często w obrazie użyty jest piksel o konkretnej wartości jasności. Pierwsza wartość histogramu ma numer 0, zaś ostatni Z_{max} (zakres maksymalny). Przy czym $Z_{max} = 2^n$, dla liczby n bitów.

Każdy punkt znajdujący się na osi poziomej pokazuje ile pikseli z przykładowego obrazu posiada wartość wskazywaną przez dany punkt. Graficzna reprezentacja wygląda następująco, iż oś X oznacza poziomy jasności (od 0 do $2^n - 1$), z kolei oś Y to procentowy udział punktów o danym poziomie jasności. Analiza histogramu dostarcza wielu niezbędnych na temat poziomów jasności, ich zakresu i częstości występowania. Z tego powodu analiza histogramu jest często niezbędna w procesie przetwarzania obrazów cyfrowych. Wiele operacji przekształcenia rozpoczyna się od wykonania histogramu obrazu. Na rysunku poniżej został pokazany obraz oraz jego histogram.



Rys. 4 Przykładowy obraz wraz z jego histogramem.

Statystycznie histogram przypomina funkcję gęstości prawdopodobieństwa $h(x)$. Pole wyznaczone przez histogram obliczamy poprzez wzór:

$$\int_0^x h(x) dx \quad (4)$$

gdzie: $h(x)$ - funkcja histogramu, dx - przyrost wartości funkcji histogramu, $h(x)dx$ – wartości kolejnych słupków funkcji histogramowej. Jeśli $dx = 1$ tak jak w przypadku obrazów cyfrowych to wartość funkcji histogramu w dowolnym punkcie x można wyznaczyć jako sumę wszystkich pikseli o danej wartości podzieloną przez ich liczbę.

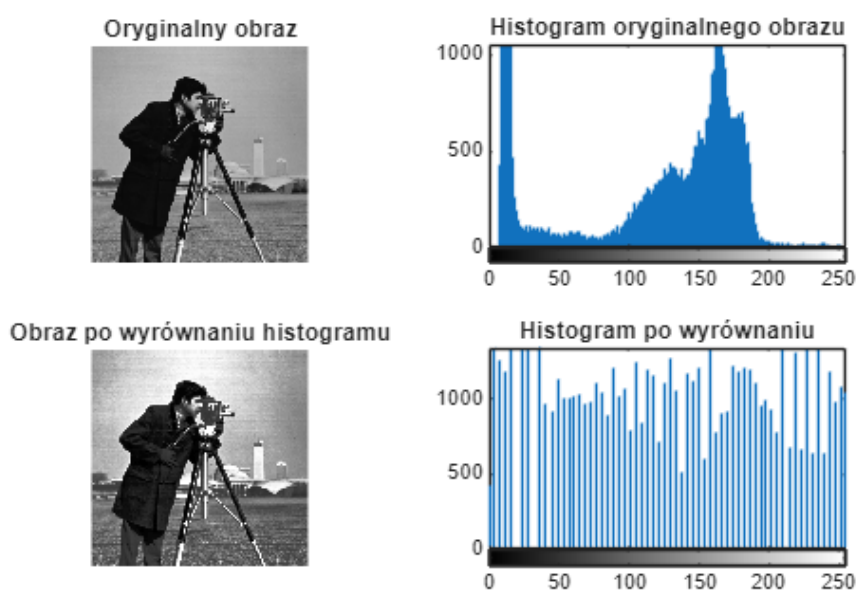
$$h(x) = \frac{SUMA(x)}{N} \quad (5)$$

gdzie: $SUMA(x)$ - liczba wszystkich pikseli o wartości x , N - liczba pikseli (np. 512x512).

Dla każdego piksela można wyznaczyć odpowiadający mu kolor. Użycie histogramu pozwala nam uzyskać informację, jakie wartości mają poszczególne piksele obrazu oraz jaka jest ilość kolorów, potrzebnych do przedstawienia danego obrazu. Tak, więc dzięki zastosowaniu histogramu mamy informacje o kontraście w obrazie. Wąski histogram oznacza, że użyto mało kolorów i wiele pikseli się powtarza. Aby poprawić kontrast, wystarczy zmodyfikować histogram. Dokonuje się tego za pomocą transformacji jednego rozkładu wartości do drugiego rozkładu wartości pikseli. Ta transformacja jest to matematyczna funkcja, a jej graficzną reprezentacją jest krzywa transformacji. Dzięki użyciu tej transformaty otrzymujemy nowy histogram o tej samej liczbie słupków, jednak ich położenie względem pierwotnego histogramu jest zmienione – są przesunięte. Używanie krzywych transformacji, daje

możliwość nie tylko poprawy kontrastu, ale także uwydatnienia pewnej cechy obrazów, która jest najbardziej interesująca w danych badaniach nad obrazem.

Jedną z operacji wykonywanych na histogramie jest jego wyrównanie. Jest to operacja, która prowadzi do polepszenia jakości danego obrazu, tak, iż łatwiej jest rozpoznać poszczególne elementy obrazu, czyli poprawia się jego czytelność. Poniższy rysunek (Rys. 2) pokazuje zmianę w obrazie po wykonaniu operacji wyrównania histogramu. Wysokie prążki, które reprezentowały odcienie szarości często powtarzające się w obrazie, rozciągnięto. Ta czynność spowodowała, że elementy opisywane przez te wartości są bardziej widoczne w obrazie. Natomiast rzadko występujące poziomy szarości, reprezentowane na histogramie przez niskie prążki, utożsamiono ze sobą. Ta operacja spowodowała, że utracono część informacji z obrazu, jednak w końcowym efekcie jest on bardziej czytelny.

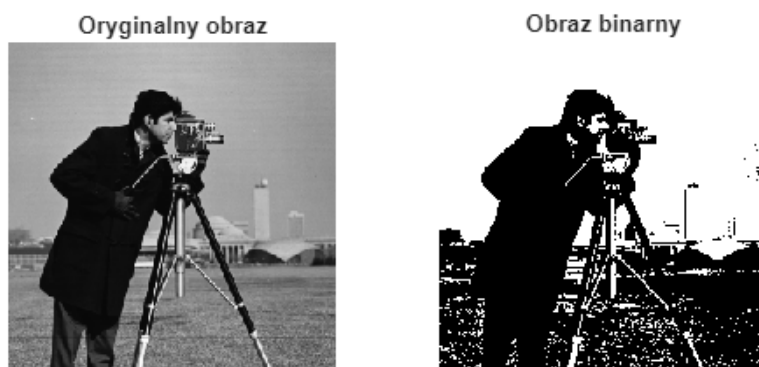


Rys. 5 Obraz po operacji wyrównania histogramu i jego histogram.

Użycie histogramu jest bardzo przydatne w procesie segmentacji obrazów, polegającej na grupowaniu obrazów pikseli w podregiony w zależności od konkretnego warunku jednorodności. Segmentacja powoduje podział obrazu na regiony, których liczba jest niższa niż liczba odcieni szarości obrazu oryginalnego. Istnieją metody segmentacji opierające się na progowaniu, czyli tworzeniu zakresów wartości, na podstawie wyliczonego histogramu. A więc możliwość wyznaczania przy użyciu histogramu rodzaju i częstotliwości pojawiania się różnych odcieni szarości, jest często wykorzystywana przy wyznaczaniu granic dla poszczególnych progów

Sposoby uzyskiwania obrazów binarnych

Binaryzacja to jedna z ważniejszych operacji dotycząca przetwarzania obrazów. Występuje jako faza wstępna w analizie obrazów oraz w procesie rozpoznawania. Dopiero obrazy binarne umożliwiają przeprowadzenie większości pomiarów i złożonych przekształceń. Najważniejszym celem, jaki można osiągnąć dzięki binaryzacji, jest uproszczenie jego struktury, poprzez zmniejszenie ilości informacji o obrazie, przy zachowaniu najważniejszych jego cech. Proces binaryzacji polega na zmianie wartości pikseli, tak, aby piksele o różnych odcieniach szarości zamienić na piksele o wartościach 0 i 1[9]. Poniżej przedstawiony został przykładowy obraz i jego postać binarna:



Rys. 6 Obraz oryginalny i jego binarna postać.

Proces binaryzacji może być przeprowadzany na wiele różnych sposobów. Najczęściej stosowana jest binaryzacja z użyciem tzw. progu.

Binaryzacja jednokryterialna polega na użyciu wartości progowej a , następnie następuje selekcja pikseli. Do jednej grupy zaliczone zostają piksele, których wartości są większe lub równe od wartości ustalonego progu, druga grupa pikseli skupia wartości poniżej progu. Przy czym używając jednej wartości progu można wyróżnić kilka rodzajów binaryzacji:

- binaryzacja z dolnym progiem

$$L'(m, n) = \begin{cases} 0 & \text{dla } L(m, n) \leq a \\ 1 & \text{dla } L(m, n) > a \end{cases} \quad (6)$$

gdzie: $L(m, n)$ – jasność punktu w obrazie źródłowym $L(m, n) \in [0, 2B - 1]$; $L'(m, n)$ – wartość odpowiedniego punktu w obrazie wynikowym $L'(m, n) \in \{0, 1\}$, a – próg binaryzacji.

- binaryzacja z górnym progiem

$$L'(m, n) = \begin{cases} 0 & \text{dla } L(m, n) \geq a \\ 1 & \text{dla } L(m, n) < a \end{cases} \quad (7)$$

- binaryzacja z podwójnym ograniczeniem

$$L'(m, n) = \begin{cases} 0 & \text{dla } L(m, n) \leq a_1 \\ 1 & \text{dla } a_1 < L(m, n) \leq a_2 \\ 0 & \text{dla } L(m, n) > a_2 \end{cases} \quad (8)$$

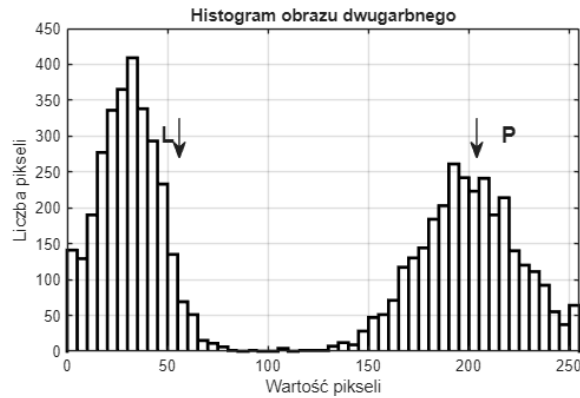
gdzie: a_1, a_2 – progi binaryzacji, $a_1 < a_2$.

- binaryzacja warunkowa (binaryzacja z histerezą)

$$L'(m, n) = \begin{cases} 0 & \text{dla } L(m, n) \leq a_1 \\ s & \text{dla } a_1 < L(m, n) \leq a_2 \\ 1 & \text{dla } L(m, n) > a_2 \end{cases} \quad (5)$$

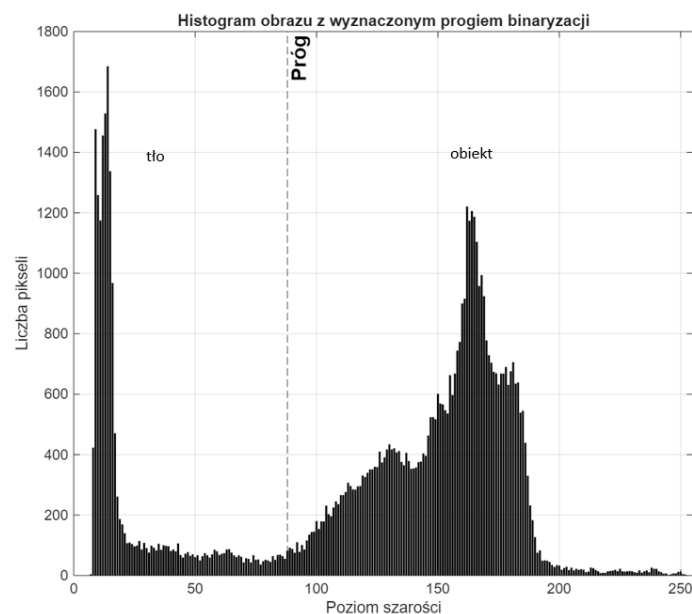
gdzie: s – wartość sąsiadujących punktów, $s \in \{0, 1\}$.

Przy procesie binaryzacji najważniejszą sprawą jest wyznaczenie odpowiedniego progu a . Najczęściej poszukiwania tej wartości dokonuje się przy użyciu histogramu. Bardzo często histogram ma postać krzywej dwugarbanej. Przykład takiego histogramu przedstawiono poniżej:



Rys. 7 Histogram obrazu, który zawiera dwie grupy pikseli.

Ten przykładowy histogram łatwo zinterpretować. Oznacza on, że w obrazie istnieją dwa rodzaje punktów, które pojawiają się bardzo często. Są to punkty ciemne – oznaczone literą *L* oraz punkty jasne oznaczone jako *P*. Aby wydzielić wybrane punkty należy zastosować filtrację z użyciem górnego lub dolnego progu. Jednak najważniejszą sprawą jest wyznaczenie wartości progu α , tak, aby występował on pośrodku „doliny” między garbami (Rys. 8).



Rys. 8 Przykład wyznaczania progu binaryzacji położonego w „dolinie” pomiędzy dwoma grupami pikseli[8].

Powyższy histogram pokazuje przynależność pikseli z obrazu do dwóch klas. Jedną klasę zawiera jasne figury położone na ciemnym tle, natomiast druga klasa oznacza obiekty złożone z ciemnych pikseli znajdujących się na tle jasnym.

Przykład. Wczytanie obrazu i analiza histogramu.

```
% Wczytanie obrazu w skali szarości
I = imread('peppers.png');
% Wyświetlenie obrazu
figure, imshow(I), title('Oryginalny obraz');
```

MATLAB automatycznie wczytuje obrazy jako macierze liczb całkowitych uint8 (dla obrazów 8-bitowych).

```
% Obliczanie histogramu
```

```
figure, imhist(I), title('Histogram obrazu');
```

Histogram można obliczyć i zwizualizować za pomocą funkcji `imhist`, która zwraca wykres pokazujący, jak często występują poszczególne poziomy szarości w obrazie.

Modyfikacja histogramu

- a) Rozciąganie histogramu (histogram stretching). Aby zwiększyć kontrast obrazu, można zastosować operację rozciągania histogramu:

```
% Normalizacja wartości pikseli
```

```
I_stretched = imadjust(I, stretchlim(I, [0.02, 0.98]), []);
```

```
% Wyświetlenie obrazu po rozciąganiu histogramu
```

```
figure, imshow(I_stretched), title('Obraz po rozciąganiu histogramu');
```

```
% Histogram po rozciąganiu
```

```
figure, imhist(I_stretched), title('Histogram po rozciąganiu');
```

Funkcja `stretchlim` ustawia nowe wartości minimalne i maksymalne, co poprawia kontrast.

- b) Wyrównywanie histogramu (histogram equalization). Wyrównywanie histogramu pozwala lepiej wykorzystać całą skalę szarości:

```
% Wyrównanie histogramu
```

```
I_eq = histeq(I);
```

```
% Wyświetlenie wyrównanego obrazu i jego histogramu
```

```
figure, imshow(I_eq), title('Obraz po wyrównaniu histogramu');
```

```
figure, imhist(I_eq), title('Histogram po wyrównaniu');
```

Segmentacja obrazu na podstawie histogramu

Histogram można wykorzystać do segmentacji obrazu poprzez progowanie.

- a) Progowanie ręczne

```
% Ustawienie progu segmentacji
```

```
threshold = 100;
```

```
BW = I > threshold;
```

```
% Wyświetlenie obrazu binarnego
```

```
figure, imshow(BW), title(['Segmentacja dla progu ', num2str(threshold)]);
```

Powyższy kod tworzy obraz binarny, gdzie piksele jaśniejsze niż próg 100 są ustawione na 1 (biały), a pozostałe na 0 (czarny)

- b) Automatyczne progowanie metodą Otsu, która automatycznie znajduje optymalny próg segmentacji:

```
% Automatyczne wyznaczenie progu metodą Otsu
```

```
level = graythresh(I);
```

```
BW_Otsu = imbinarize(I, level);
```

```
% Wyświetlenie obrazu binarnego po segmentacji metodą Otsu
```

```
figure, imshow(BW_Otsu), title('Segmentacja metodą Otsu');
```

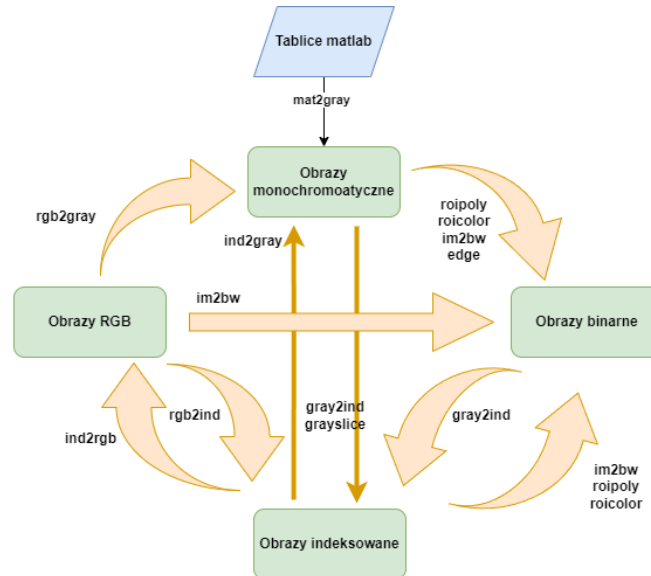
Funkcja `graythresh` zwraca wartość progu, która minimalizuje wariancję między obiektami a tłem.

4. Zadania do samodzielnego rozwiązania

Zaproponowane rozwiązania poniższych zadań należy umieścić w sprawozdaniu z laboratorium. W zadaniach należy wykorzystać przykładowe obrazy z bazy danych odcisków palców i tęczy, które udostępnione są w materiałach do zajęć jak również dostępne są na stronie Neurotechnology pod adresem: <https://www.neurotechnology.com/download.html#databases> (data dostępu 23.02.2025r).

Zadanie 1. Konwersja obrazu na różne formaty przy użyciu MATLAB i Image Processing Toolbox

Użyj funkcji dostępnych w bibliotece IPT w MATLAB-ie do przekształcenia obrazu. Na Rys. 9 przedstawiono schemat przepływu i konwersji różnych typów obrazów w MATLAB-ie, z wykorzystaniem funkcji dostępnych w IPT.



Rys. 9. schemat przepływu i konwersji różnych typów obrazów w MATLAB-ie.

W zadaniu:

- Wczytaj dowolny obraz kolorowy (RGB) za pomocą funkcji `imread`.
- Dokonać konwersji na obraz monochromatyczny.
- Dokonać binaryzacji obrazu.
- Przekształcić obraz do formatu indeksowanego.
- Zwizualizować osobno kanały obrazu RGB.
- Zapisać wszystkie przekształcone obrazy do plików (`imwrite`).

Ponadto odpowiedź na poniższe pytania:

- Jak zmienia się zakres wartości pikseli w kolejnych przekształceniach?
- Jaka jest różnica w zajmowanej pamięci między różnymi formatami obrazów?
- Jakie mogą być zastosowania obrazów indeksowanych i binarnych w praktyce?

Zadanie 2. Histogram i operacje na histogramie

W Matlab zaproponuj program, który pozwoli na:

- Wczytanie obrazu w skali szarości (`imread` → `rgb2gray` jeśli potrzeba).
- Obliczenie i wyświetlenie histogramu obrazu (`imhist`).
- Wykonaj rozciąganie histogramu (`imadjust` + `stretchlim`) i porównaj obraz przed i po operacji.
- Wykonaj wyrównanie histogramu (`histeq`) i porównaj wynik z oryginalnym obrazem oraz z histogramem rozciągniętym.

- Wyświetl histogramy wszystkich przekształconych obrazów.

Pytania do analizy:

- Jaka jest różnica między rozciąganiem a wyrównywaniem histogramu?
- Która metoda daje lepsze wyniki poprawy kontrastu?
- Jakie są potencjalne zastosowania tych technik?

Zadanie 3. Segmentacja obrazu na podstawie histogramu

W Matlab zaproponuj program, który pozwoli na:

- Wczytanie obrazu w skali szarości.
- Obliczenie histogramu obrazu (imhist) i znalezienie potencjalnego progu segmentacji.
- Przeprowadź ręczną segmentację obrazu, stosując różne wartości progowe ($BW = I > \text{threshold}$).
- Zastosuj automatyczną segmentację metodą Otsu (graythresh + imbinarize).
- Porównaj efekty różnych progowań, wyświetlając wyniki obok siebie.

Pytania do analizy:

- Jak dobrać optymalny próg segmentacji dla różnych obrazów?
- Kiedy metoda Otsu jest skuteczniejsza niż segmentacja ręczna?
- Jakie mogą być praktyczne zastosowania binaryzacji w analizie obrazów?

Zadanie 4. Detekcja obiektów na obrazie binarnym

W Matlab zaproponuj program, w którym:

- Wykonasz segmentację obrazu metodą Otsu i uzyskasz obraz binarny.
- Zidentyfikuj obiekty na obrazie binarnym za pomocą funkcji bwlabel lub regionprops.
- Wyznacz podstawowe właściwości wykrytych obiektów (np. powierzchnię, obwód).
- Narysuj prostokątne obramowania wokół wykrytych obiektów (rectangle).
- Zwizualizuj oryginalny obraz z nałożonymi obramowaniami.

Pytania do analizy:

- Jakie problemy mogą wystąpić przy detekcji obiektów w zależności od jakości segmentacji?
- Jak można poprawić dokładność detekcji?
- Jakie są praktyczne zastosowania tej metody (np. w biomedycynie, kontroli jakości)?

Zadanie 5. Segmentacja wieloprogowa

W Matlab zaproponuj program, w którym:

- Wczytasz obraz w skali szarości.
- Obliczysz histogram obrazu i znajdziesz dwa optymalne progi metodą Otsu (multithresh).
- Utwórz obraz segmentowany na trzy klasy (imquantize).
- Zwizualizuj wynik segmentacji z użyciem różnych kolorów (label2rgb).
- Porównaj segmentację wieloprogową z binaryzacją jednokryterialną.

Pytania do analizy:

- W jakich przypadkach segmentacja wieloprogowa daje lepsze wyniki niż binaryzacja?
- Jakie są jej ograniczenia?
- Jak można poprawić jakość segmentacji?