**1. W programie Matlab należy wczytać i wyświetlić po jednym przykładzie pliku graficznego z rozszerzeniem: .jpeg , .bmp, .tiff, animowny gif, .png:**

**a) dla poszczególnych rozszerzeń dokładnie opisać funkcję *imread* (zarówno parametry wejściowe jak i wyjściowe)**



Rys. 1. Wyświetlenie obrazów w formatach jpeg, bmp, tiff, png, giff

%% Wczytanie obrazów

obraz\_jpg=imread('miasto.jpg');

obraz\_bmp=imread('czolg.bmp');

obraz\_tiff=imread('napis.tiff');

obraz\_png=imread('lew.png');

[A, map]=imread('pingwin.gif',1:8);

%%Wyświetlenie obrazów

figure(1)

subplot(2,3,1)

imshow(obraz\_jpg);

title('Obraz JPEG')

subplot(2,3,2)

imshow(obraz\_bmp);

title('Obraz BMP')

subplot(2,3,3)

imshow(obraz\_tiff);

title('Obraz TIFF')

subplot(2,3,4)

imshow(obraz\_png);

title('Obraz PNG')

subplot(2,3,5)

for i=1:8 %wyświetlenie 8 klatek GIF

imshow(A(:,:,:,i),map)

end

title('Obraz GIF');

*Imread* jest to funkcja, która służy do wczytywania plików graficznych w programie Matlab. Danymi wejściowymi jest nazwa wczytywanego pliku i jego rozszerzenie, natomiast parametry wyjściowe tej funkcji to macierz dwuwymiarowa (obraz indeksowany). Jeden piksel jest zawarty w danym elemencie tablicy poprzez określenie jego położenia za pomocą numeru wiersza i kolumny, natomiast wartości elementów obrazu reprezentują nasycenie pikseli w skali od 0 do 1.

**b) dla wybranego obrazu (wybrane rozszerzenie) opisać dostępne informacje o pliku graficznym: polecenie *imfinfo*. Następnie wybraną cechę/parametr wykorzystać w dowolnym innym skrypcie**

%% Wyświetlenie informacji o obrazach

info\_jpg=imfinfo('miasto.jpg');

info\_bmp=imfinfo('czolg.bmp');

info\_tif=imfinfo('napis.tiff');

info\_png=imfinfo('lew.png');

imfinfo('pingwin.gif');

info\_jpg =

Filename: 'C:\Users\lenovo\Desktop\Projekt obrazy 2\miasto.jpg'

FileModDate: '09-Jun-2017 22:15:53'

FileSize: 3008694

Format: 'jpg'

FormatVersion: ''

Width: 5120

Height: 2880

BitDepth: 24

ColorType: 'truecolor'

FormatSignature: ''

NumberOfSamples: 3

CodingMethod: 'Huffman'

CodingProcess: 'Sequential'

Comment: {}

Make: 'NIKON CORPORATION'

Model: 'NIKON D3200'

XResolution: 240

YResolution: 240

ResolutionUnit: 'Centimeter'

Software: 'Adobe Photoshop Lightroom 6.4 (Windows)'

DateTime: '2016:02:28 20:25:16'

Artist: 'Best of Cinque Terre team'

Copyright: 'https://www.bestofcinqueterre.com'

DigitalCamera: [1×1 struct]

GPSInfo: [1×1 struct]

ExifThumbnail: [1×1 struct]

Przy pomocy funkcji *imfinfo* możemy odczytać takie informacje o obrazie jak ścieżkę jego lokalizacji, datę zapisania, rozmiar pliku, format, rozdzielczość, datę utworzenie czy też model urządzenia za pomocą którego obraz został zrobiony.

info=imfinfo('miasto.jpg')

if info.Height < 1080

afterInfo = imrotate(obraz\_jpg, 90);

else

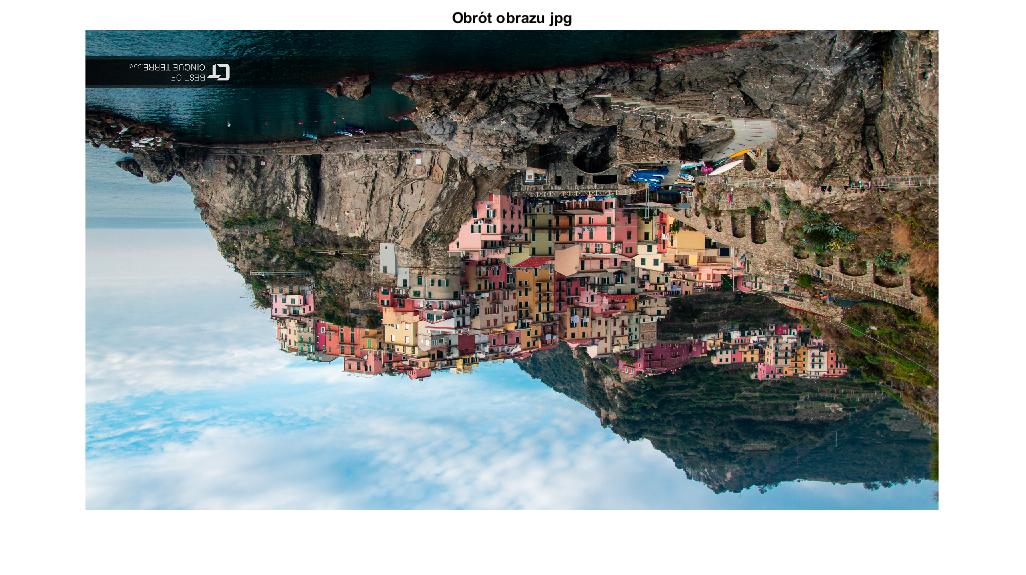
afterInfo = imrotate(obraz\_jpg, 180);

end

figure (1)

imshow(afterInfo)

title('Obrót obrazu jpg')



Rys. 2. Obrót obrazu jpg

Powyższy skrypt realizuje obrót obrazu w formacie jpg w zależności od jego wysokości. Jeśli wysokość jest mniejsza niż 1080 pikseli, obraz zostaje obrócony o 90°, w przeciwnym razie obraz jest obracany o 180°.

**c) dla wybranego obrazu porównać funkcje służące wyświetlaniu obrazu: *imshow*, *image*, *imagsc*, *warp* (omówić i pokazać różnice)**

%% Porównanie funkcji

figure(2)

subplot(2,2,1)

imshow(obraz\_jpg);

title('Funkcja imshow')

subplot(2,2,2)

image(obraz\_jpg);

title('Funkcja image')

subplot(2,2,3)

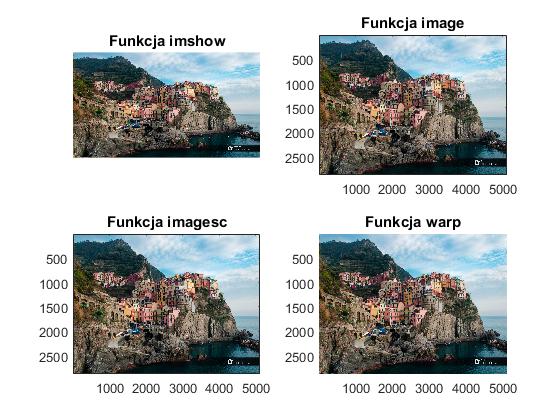
imagesc(obraz\_jpg);

title('Funkcja imagesc')

subplot(2,2,4)

warp(obraz\_jpg);

title('Funkcja warp')



Rys. 3. Porównanie wyświetlenia obrazu jpg przy użyciu funkcji imshow, image, imagesc, warp

Funkcja *imshow* wyświetla obraz oraz go optymalizuje. Kolejna funkcja – *image* działa w w ten sam sposób co *imshow,* z tym, że obraz wyświetlany jest w postaci macierzowej. Funkcja *imagesc* pozwala uzyskać najlepszą jakość poprzez wydobywanie mapy kolorów. Funkcja *warp* wyświetla obraz w postaci tekstury.

**d) wybrany obraz zapisać w trzech innych rozszerzeniach – podając i opisując istotne parametry z punktu widzenia procedury zapisu w tym również parametry domyślne - (funkcja *imwrite* – charakterystyka funkcji wraz z możliwymi parametrami)**

%% Zapisanie JPG w innych rozszerzeniach

imwrite(obraz\_jpg,'zJPGdoBMP.bmp'); %zapis do bmp

imwrite(obraz\_jpg,'zJPGdoTIF.tiff'); %zapis do tiff

imwrite(obraz\_jpg,'zJPGdoPNG.png'); %zapis do png

Funkcja imwrite umożliwia konwersję obrazu do innego formatu. Jako pierwszy argument należy podać nazwę obrazu, którego format chcemy zmienić. Drugim argumentem jest nazwa nowego obrazu wraz z pożądanym rozszerzeniem. Plik zapisywany jest w aktualnym katalogu roboczym. Głębia obrazu wyjściowego zależy od typu danych i formatu pliku. *Imwrite* pozwala na zapis danych do pliku jpg, png, gif, tiff a także bmp. *Imwrite* może zmienić rozmiar pliku wyjściowego.

**2. Dokonać nieliniowego przekształcenie obrazu (zarówno dla obrazu barwnego jak i obrazu w odcieniach szarości) - (w wyciąganiu wniosków pomocny będzie podgląd z przestrzeni roboczej Matlaba wartości macierzy reprezentujących dane obrazowe):**

**a) pierwiastkowanie (sqrt)**

%% Nieliniowe przekształcenie obrazu

%%Pierwiastkowanie

o\_rgb = obraz\_jpg;

o\_drgb = double(o\_rgb)/255;

o\_gray = rgb2gray(obraz\_jpg);

o\_dgray = double(o\_gray)/255;

figure(3)

subplot(2,2,1)

imshow(o\_rgb);

title('Obraz barwny')

subplot(2,2,2)

imshow(sqrt(o\_drgb));

title('Obr. barwny spierwiastkowany');

subplot(2,2,3)

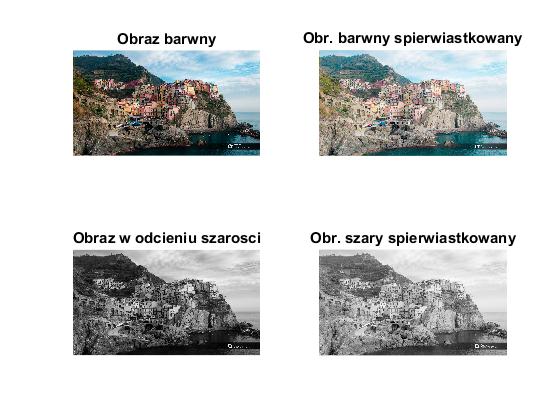
imshow(o\_gray);

title('Obraz w odcieniu szarosci')

subplot(2,2,4)

imshow(sqrt(o\_dgray));

title('Obr. szary spierwiastkowany');



Rys. 4. Obraz barwy oraz skali szarości przed i po operacji pierwiastkowania

Obraz barwny RGB należało przekonwertować do obrazu w skali szarości za pomocą funkcji *rgb2gray*. Następnie należało przekonwertować obraz z klasy unit8 (8-bitowa zmienna bez znaku) do klasy double (64-bitowa liczba zmiennoprzecinkowa. Do pierwiastkowania obrazu wykorzystano funkcje *sqrt*. Pierwiastkowanie powoduje zmniejszenie intensywności obrazu.

**b) potęgowanie (obraz.^2)**

%% Potęgowanie

figure(4)

subplot(2,2,1)

imshow(o\_rgb);

title('Obraz barwny')

subplot(2,2,2)

imshow(o\_drgb.^2);

title('Obr. barwny potęgowany');

subplot(2,2,3)

imshow(o\_gray);

title('Obraz w odcieniu szarosci')

subplot(2,2,4)

imshow(o\_dgray.^2);

title('Obr. szary potęgowany');



Rys. 5. Obraz barwy oraz skali szarości przed i po operacji potęgowania

Operacja potęgowania powoduje przyciemnienie obrazu poprzez zwiększenie intensywności odcieni.

**c) podwojenie wartości.**

%% Podwojenie

figure(5)

subplot(2,2,1)

imshow(o\_rgb);

title('Obraz barwny')

subplot(2,2,2)

imshow(o\_drgb.\*2);

title('Obr. barwny podwojony');

subplot(2,2,3)

imshow(o\_gray);

title('Obraz w odcnieu szarosci')

subplot(2,2,4)

imshow(o\_dgray.\*2);

title('Obr. szary podwojony');



Rys. 6. Obraz barwy oraz skali szarości przed i po operacji podwojenie wartości

Operacja podwojenia wartości powoduje rozjaśnienie obrazu. Wartości parametrów obrazów zostały przemnożone przez liczbę 2.

**3. Dokonać konwersji obrazu barwnego RGB do modelu: HSV, NTSC, YCbCr (*rgb2hsv*, *hsv2rgb*, *rgb2ntsc*, *ntsc2rgb*, *rgb2ycbcr*, *ycbcr2rgb*) dokładnie opisać funkcje - składnię, typy danych wejściowych i wyjściowych, przedstawić obraz wynikowy oraz jego składowe)**

**Konwersja z RGB to HSV:**

%% Konwersja obrazów

%%RGB na hsv

rgb\_hsv=rgb2hsv(o\_rgb);

hsv\_rgb=hsv2rgb(rgb\_hsv);

figure(6)

subplot(1,2,1)

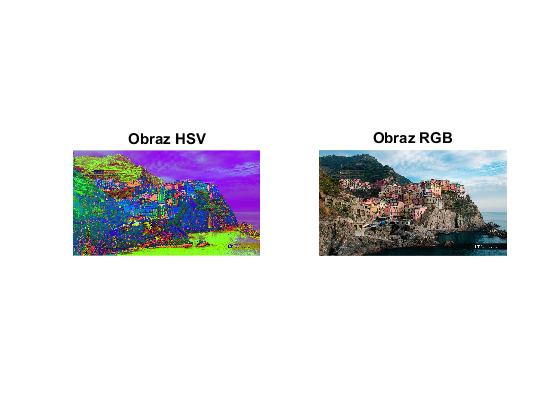
imshow(rgb\_hsv);

title('Obraz HSV')

subplot(1,2,2)

imshow(hsv\_rgb);

title('Obraz RGB');



Rys. 7. Konwersja obrazu z RGB to HSV

Model HSV (Hue, Saturation, Value) jest modelem dla kolorów postrzeganych przez człowieka. Model jest przedstawiony jako odwrócony stożek. Każdej barwie przyporządkowuje punkt w przestrzeni trójwymiarowej definiowanej przez trzy składowe: H (odcień), S (nasycenie), V (jasność). Wartość H mieści się w zakresie od 0 do 360°, a wartości S i V od 0 do 1. Mapa kolorów RGB zamieniana jest na mapę kolorów HSV.

**Konwersja z RGB do NTSC:**

%%RGB na NTSC

rgb\_ntsc=rgb2ntsc(o\_rgb);

ntsc\_rgb=ntsc2rgb(rgb\_ntsc);

figure(7)

subplot(1,2,1)

imshow(rgb\_ntsc);

title('Obraz NTSC')

subplot(1,2,2)

imshow(ntsc\_rgb);

title('Obraz RGB');



Rys. 8. Konwersja obrazu z RGB do NTSC

NTSC (**N**ational **T**elevision **S**ystem **C**ommittee) jest systemem stosowanym w amerykańskiej telewizji analogowej. Fala podnośna przenosi w nim informację o 2 składowych barwy: czerwonym i zielonym. Obraz został nasycony kolorem czerwonym i występują na nim małe detale w kolorze zielonym.

**Konwersja z RGB do YCbCr:**

%%RGB na YCbCr

rgb\_ycbcr=rgb2ycbcr(o\_rgb);

ycbcr\_rgb=ycbcr2rgb(rgb\_ycbcr);

figure(8)

subplot(1,2,1)

imshow(rgb\_ycbcr);

title('Obraz YCbCr')

subplot(1,2,2)

imshow(ycbcr\_rgb);

title('Obraz RGB');



Rys. 9. Konwersja obrazu z RGB do YCbCr

Model YCbCr stosowany jest w formacie jpeg. Wykorzystuje 3 składowe: Y- składową luminancji, Cb- składową różnicy chrominancji Y-B (różnica pomiędzy luminacją a niebieskim) oraz Cr- skladowa chrominancji Y-R (różnica między luminacją a czerwonym). Ten typ modelu rozdziela luminację i chrominancję oraz wykorzystuje małą korelację składowych. Model RGB wykazuje dużą korelację składowych i jego kompresja jest nieefektywna.

**4. Dla dowolnego obrazu barwnego napisać skrypt realizujący wyświetlanie we współrzędnych 3D rozkładu barw podstawowych R, G, B (przy zachowaniu odpowiedniej kolorystyki). Dodatkowo dla wybranego kanału (R lub G lub B) dokonać uproszczenia polegającego na wprowadzeniu płaszczyzny kryterialnej o danej wartości (Z – nasycenie) redukującą pewne wartości nasycenia wybranej barwy podstawowej.**

%% Wyświetlenie kolorów we współrzędnych 3D

%%Barwy

obrazR=obraz\_jpg(:,:,1); %Kanał czerwony

obrazG=obraz\_jpg(:,:,2); %Kanał zielony

obrazB=obraz\_jpg(:,:,3); %Kanał niebieski

[w,k]=size(obrazR);

%Wyswietlanie kanalu czerwonego

r=0:1:255;

rr=r';

z=zeros(256,3);

z(:,1)=rr;

red=z/255;

%Wyswietlanie kanalu zielonego

r2=0:1:255;

rr2=r2';

z2=zeros(256,3);

z2(:,2)=rr2;

green=z2/255;

%Wyswietlanie kanalu niebieskiego

r3=0:1:255;

rr3=r3';

z3=zeros(256,3);

z3(:,3)=rr3;

blue=z3/255;

figure(9)

mesh(1:k,w:-1:1,obrazR,'facecolor','texturemap','edgecolor','none','cdatamapping','direct');

colormap(red);

view(3);

colorbar('vert');

title('Wykres 3D-kanal czerwony');

xlabel('Kolumna');

ylabel('Wiersz');

zlabel('Natężenie');

figure(10)

mesh(1:k,w:-1:1,obrazG,'facecolor','texturemap','edgecolor','none','cdatamapping','direct');

colormap(green);

view(3);

colorbar('vert');

title('Wykres 3D-kanal zielony');

xlabel('Kolumna');

ylabel('Wiersz');

zlabel('Natężenie');

figure(11)

mesh(1:k,w:-1:1,obrazB,'facecolor','texturemap','edgecolor','none','cdatamapping','direct');

colormap(blue);

view(3);

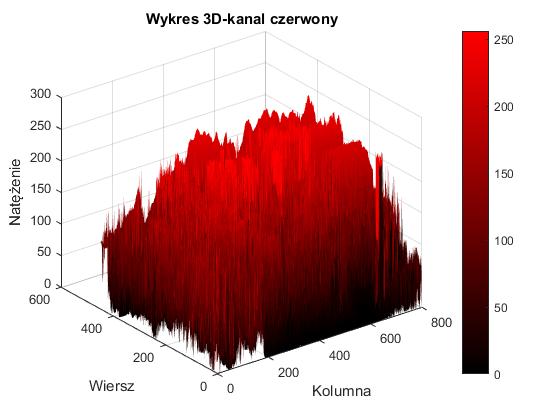
colorbar('vert');

title('Wykres 3D-kanal niebieski');

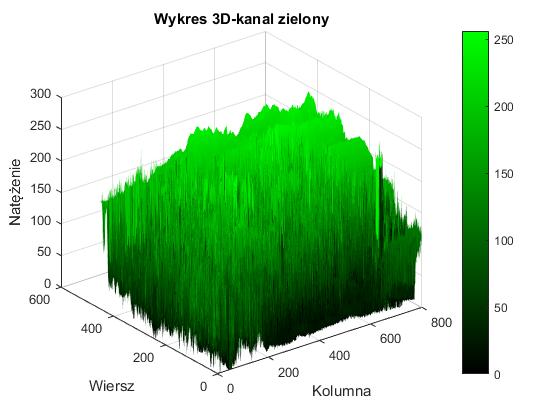
xlabel('Kolumna');

ylabel('Wiersz');

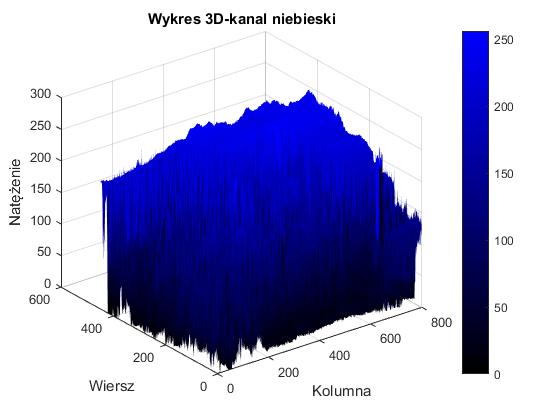
zlabel('Natężenie');



Rys. 10. Przedstawienie modelu 3D dla kanału czerwonego



Rys. 11. Przedstawienie modelu 3D dla kanału zielonego



Rys. 12. Przedstawienie modelu 3D dla kanału niebieskiego

Wykresy przedstawiają przestrzenny rozkład barw składowych RGB. Do wygenerowania wykresu wykorzystano funkcję *mesh*. Na poszczególnych osiach przedstawiono kolejno liczbę kolumn, liczbę wierszy oraz kanał z danym kolorem. Na podstawie wykresu 3D można określić w jakim miejscu na obrazie występuje nasycenie koloru o danej wartości.

**a) przedstawić histogramy R, G, B (odpowiedni dla danych kolorów colorbar)**

%% Histogramy

figure(12)

dane\_hist=imhist(obrazR);

colormap(red);

imhist(obrazR);

colorbar('location','southoutside');

grid on;

title('Histogram - kanal czerwony');

figure(13)

dane\_hist=imhist(obrazG);

colormap(green);

imhist(obrazG);

colorbar('location','southoutside');

grid on;

title('Histogram - kanal zielony');

figure(14)

dane\_hist=imhist(obrazB);

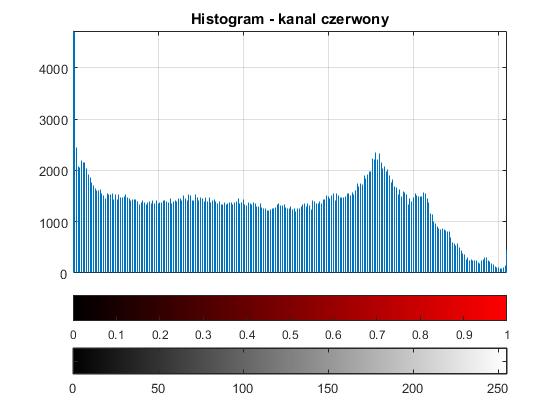
colormap(blue);

imhist(obrazB);

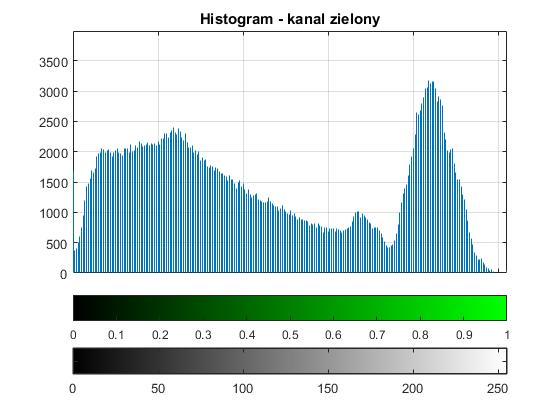
colorbar('location','southoutside');

grid on;

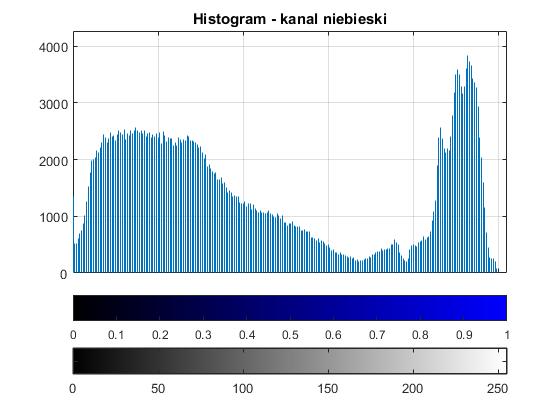
title('Histogram - kanal niebieski');



Rys. 13. Histogram dla kanału czerwonego



Rys. 14. Histogram dla kanału zielonego



Rys. 15. Histogram dla kanału niebieskiego

Histogram pokazuje ilość pikseli w zależności od poziomu jasności. Na osi poziomej odłożone są jasności koloru w zakresie od 0 do 1, a na osi pionowej liczba pikseli dla poszczególnych poziomów jasności. Na osi poziomej przedstawiono również zależność liczby pikseli od stopnia szarości na obrazie. Histogramy są pomocne podczas określania progu binaryzacji.

**b) wyświetlić profil barwny RGB: wzdłuż wybranego odcinka na obrazie, w którym dochodzi do dużych skoków wartości składowych R, G, B (np. przedmiot na silnie kontrastowym tle). Wnioski powinny obejmować dwojakie podejście do sposobu wyświetlania profilu (bez i na podstawie danych zwracanych przez funkcję)**

%% Wykreslenie profilu barwnego RGB wzdłuż wybranego odcinka

figure(15)

imshow(obraz\_jpg);

c=improfile(obraz\_jpg,[730 740],[275 325]);

line([730 740],[275 325],'Color','r','Linewidth',5);

figure(16)

plot(c(:,:,1),'r');

hold on;

plot(c(:,:,2),'g');

hold on;

plot(c(:,:,3),'b');

hold on;

title('Profile barwne RGB wzdłuż danej linii');

grid on;

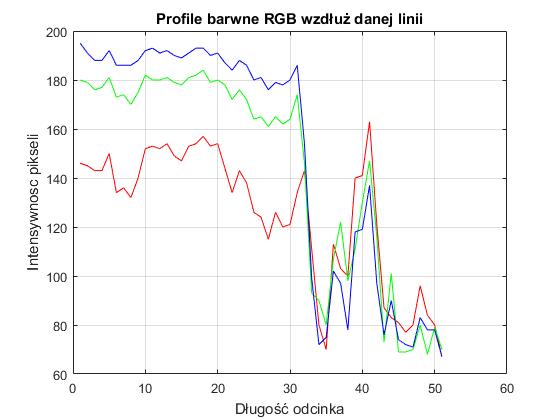
xlabel('Długość odcinka');

ylabel('Intensywnosc pikseli');



Rys. 16. Odcinek zaznaczony na obrazie

Linia przedstawiona na rys. 16. Została przeprowadzona na obrazie w miejscu, gdzie dochodzi do dużej zmiany wartości RGB- jasne tło i ciemny obiekt.



Rys. 17. Profile barwne wzdłuż wybranego odcinka

W początkowej fazie przebiegu odcinka intensywność kolorów jest największa i dominuje kolor niebieski (jasny odcień wody), a następnie intensywność kolorów znacznie spada, ponieważ obiektem jest ciemna skała.. Wartości na poszczególnych kanałach są podobne i mieszczą się w granicach 70-80, co wskazuje na kolor szary.

**c) zmodyfikować wybrany kanał barwny (np. poprzez odjęcie, dodanie, mnożenie itd. elementów w macierzy), a następnie za pomocą funkcji cat scalić macierze; wyświetlić obraz**

%% Modyfikacja wybranego kanału barwnego

ObrazR=obraz\_jpg(:,:,1);

ObrazG=obraz\_jpg(:,:,2);

ObrazB=obraz\_jpg(:,:,3)/5;

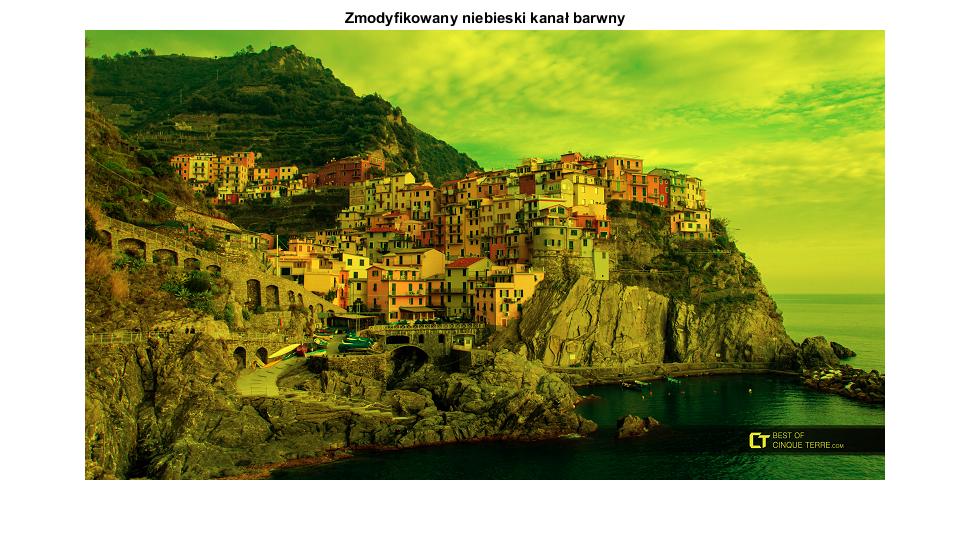
obraz\_mod=cat(3,ObrazR,ObrazG,ObrazB);

figure(17)

imshow(obraz\_mod);

title('Zmodyfikowany niebieski kanał barwny');

Na Rys. 18. Przedstawiono modyfikację obrazu poprzez podzielenie wartości kanału niebieskiego przez 5. Ograniczyło to występowanie barwy niebieskiej, kosztem wzrostu intensywności barwy zielonej. Woda, chmury oraz niebo przybrały zielony odcień.



Rys. 18. Obraz barwny po modyfikacji kanału niebieskiego

**d) pseudokolorowanie (3 dowolne palety kolorów) dla 3 wartości: 16, 128, 255 kolorów**

%% Pseudokolorowanie

[ObrazRGB]=imread('miasto.jpg');

[x1,map1]=rgb2ind(ObrazRGB,16);%16 ilość poziomów danej palety

[x2,map2]=rgb2ind(ObrazRGB,128);%128 ilość poziomów danej palety

[x3,map3]=rgb2ind(ObrazRGB,255);%255 ilość poziomów danej palety

figure(18)

subplot(1,3,1);

imshow(x1,colormap(prism));

title('16 kolorow');

subplot(1,3,2);

imshow(x2,colormap(prism));

title('128 kolorow');

subplot(1,3,3);

imshow(x3,colormap(prism));

title('255 kolorow');

figure(19)

subplot(1,3,1);

imshow(x1,colormap(copper));

title('16 kolorow');

subplot(1,3,2);

imshow(x2,colormap(copper));

title('128 kolorow');

subplot(1,3,3);

imshow(x3,colormap(copper));

title('255 kolorow');

figure(20)

subplot(1,3,1);

imshow(x1,colormap(jet));

title('16 kolorow');

subplot(1,3,2);

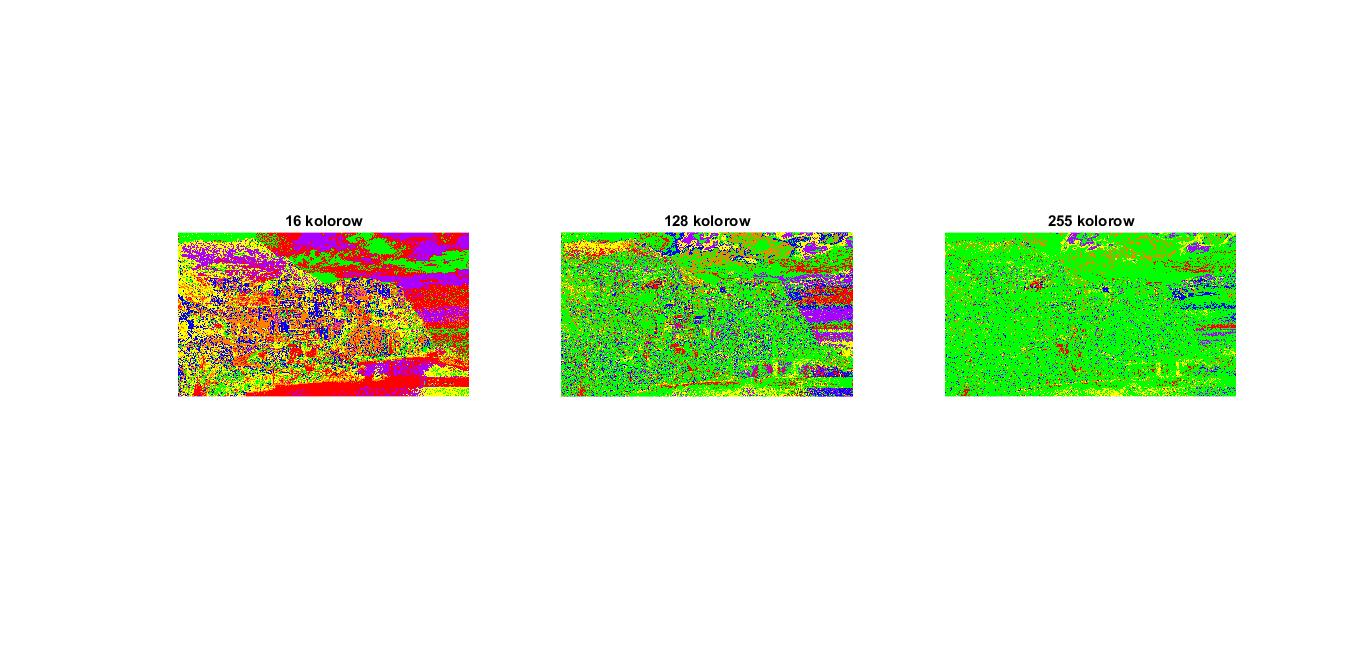
imshow(x2,colormap(jet));

title('128 kolorow');

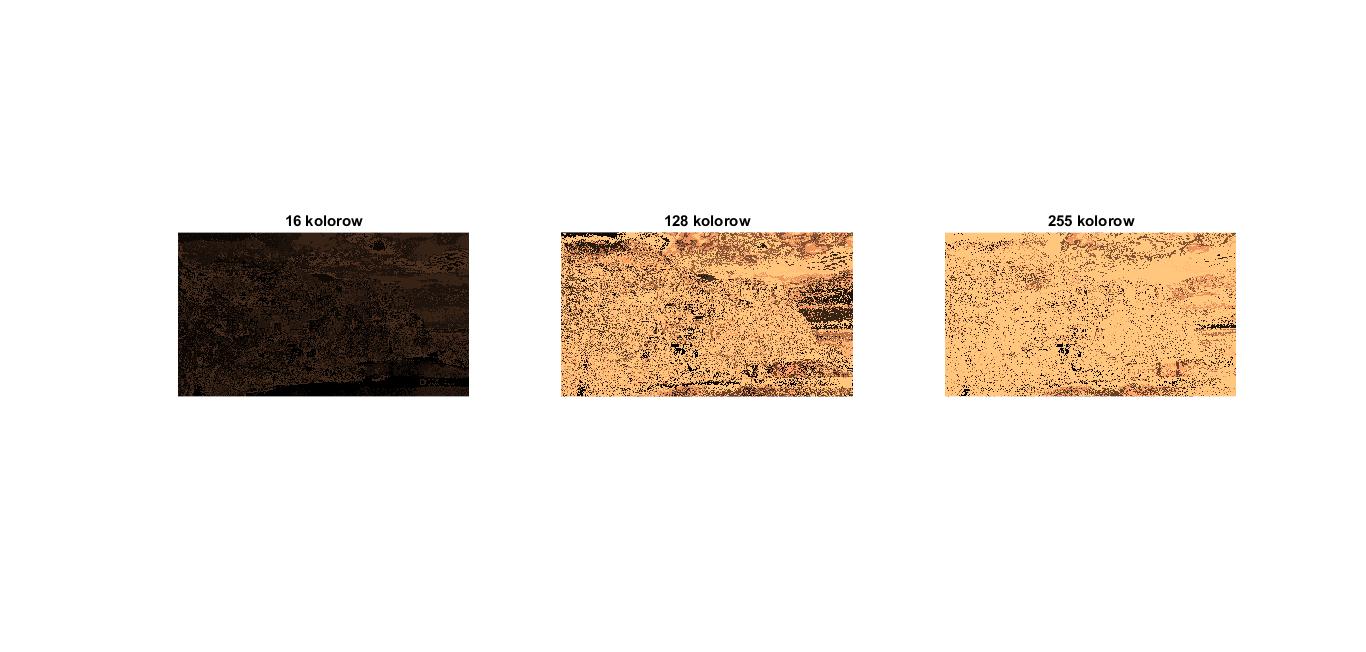
subplot(1,3,3);

imshow(x3,colormap(jet));

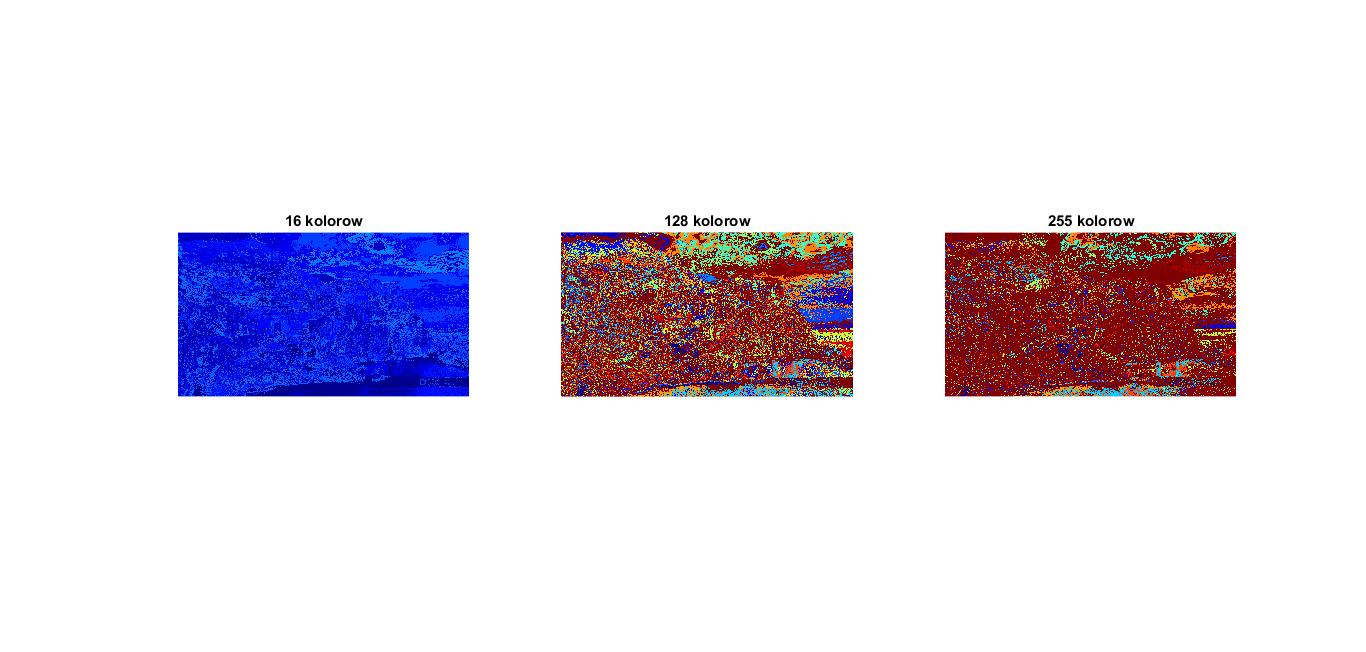
title('255 kolorow');



Rys. 19. Użycie palety kolorów prism



Rys. 20. Użycie palety kolorów copper



Rys. 21. Użycie palety kolorów jet

Do pseudokolorowania wykorzystano 3 palety barw: prism, copper i jet. Wykresy przedstawiono na 3 subplotach kolejno dla 16, 128 i 255 kolorów w poszczególnej palecie barw. Z analizy obrazów widać, że wraz ze wzrostem liczby kolorów w palecie barw obraz staje się coraz bardziej zniekształcony. Nawet dla 8 kolorów w palecie nie da się rozszyfrować początkowego obrazu RGB.

1. **Przedstawić działanie poszczególnych funkcji wraz z opisem-składnią (obraz barwny, obraz w odcieniach szarości):**
2. **imrotate (dla wybranego kąta obrotu zbadać metody interpolacyjne 'nearest','bilinear','bicubic' oraz opcje związane z wyświetlaniem obrazu 'croop', 'loose')**

%% Obracanie obrazu

J11=imrotate(o\_rgb,-45,'nearest','crop');

J12=imrotate(o\_gray,-45,'nearest','loose');

J21=imrotate(o\_rgb,-45,'bilinear','crop');

J22=imrotate(o\_gray,-45,'bilinear','loose');

J31=imrotate(o\_rgb,-45,'bicubic','crop');

J32=imrotate(o\_gray,-45,'bicubic','loose');

figure(21)

subplot(1,2,1)

imshow(J11)

title('Obraz rgb obrócony o 45°')

subplot(1,2,2)

imshow(J12)

title('Obraz szary obrócony o 45°')

figure(22)

subplot(1,2,1)

imshow(J21)

title('Obraz rgb obrócony o 45°')

subplot(1,2,2)

imshow(J22)

title('Obraz szary obrócony o 45°')

figure(23)

subplot(1,2,1)

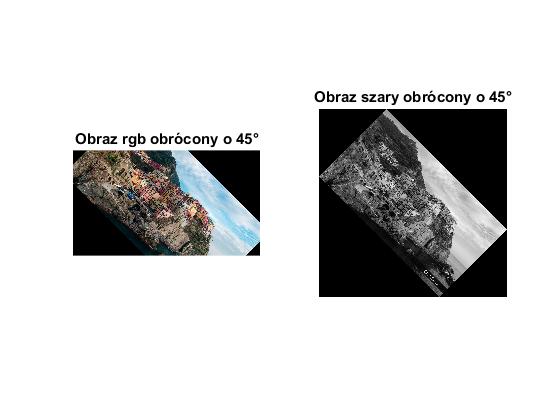
imshow(J31)

title('Obraz rgb obrócony o 45°')

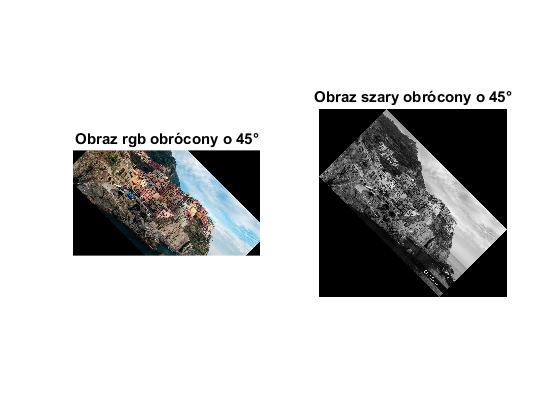
subplot(1,2,2)

imshow(J32)

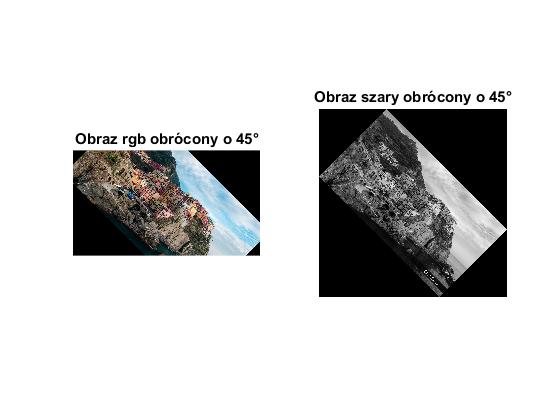
title('Obraz szary obrócony o 45°')



Rys. 22. Obrót obrazów z wykorzystanie interpolacji najbliższego sąsiada



Rys. 23. Obrót obrazów z wykorzystanie interpolacji bilinear



Rys. 24. Obrót obrazów z wykorzystanie interpolacji bicubic

Obrazy zostały obrócone funkcją *imrotate* obrócone o kąt 45° zgodnie z ruchem wskazówek zegara. Jako argumenty funkcji podaje się nazwę obrazu, kąt obrotu w stopniach, sposób interpolacji oraz rozmiar obrazu wyjściowego. Obraz barwnego zostało obrócony z parametrem *crop*, co oznacza, że na wyjściu obraz został przycięty do rozmiaru obrazu początkowego, a więc cześć pikseli obrazu podstawowego została utracona. Parametr *loose* zwiększa rozmiar obrazu wynikowego, ale nie tracimy wtedy pikseli obrazu podstawowego podczas obrotu.

Interpolacja metodą najbliższego sąsiada (nearest) w przypadku powiększania wiernie kopiuje najbliższe piksele. W przypadku pomniejszania obrazu pomija niektóre piksele. Jest to najprostsza metoda i wymaga najmniej mocy obliczeniowej od komputera.

Interpolacja dwuliniowa (bilinear) daje lepszy efekt niż metoda najbliższego sąsiada. Piksele są powielane lub redukowane z uwzględnieniem koloru czterech najbliższych pikseli.

Interpolacja dwusześcienna (bicubic) daje po transformacji najlepsze efekty a obraz najbardziej przypomina obraz początkowy. W tej metodzie uwzględnianie są kolory ośmiu pikseli sąsiadujących z danym pikselem.

1. **maketform (+imtransform) - na podstawie kilku przekształceń scharakteryzować parametry funkcji: trans=maketform('WSTAW',[a1 a2; a3 a4; a5 a6])**

%% Maketfrom

T = maketform('affine',[.5 0 0; .5 2 0; 0 0 1]);

tformfwd([10 20],T);

T1 = imtransform(o\_gray,T);

figure(24)

imshow(T1)

Za pomocą funkcji maketform i imtransform można utworzyć transformację obrazu, dzięki której sprawia on wrażenie obrazu przestrzennego.



Rys. 25. Transformacja obrazu metodą maketform

1. **imresize (przeanalizować zmianę rozdzielczości obrazu powiększanie/ \pomniejszenia obrazu dla trzech metod interpolacyjnych 'nearest','bilinear','bicubic')**

%% Imersize

IS12=imresize(o\_rgb,0.5,'nearest');

IS22=imresize(o\_rgb,0.5,'bilinear');

IS32=imresize(o\_rgb,0.5,'bicubic');

IS11=imresize(o\_gray,1.5,'nearest');

IS21=imresize(o\_gray,1.5,'bilinear');

IS31=imresize(o\_gray,1.5,'bicubic');

figure(25)

subplot(3,1,1)

imshow(IS11)

title('Obraz szary powiększony')

subplot(3,1,2)

imshow(IS21)

title('Obraz szary powiększony')

subplot(3,1,3)

imshow(IS31)

title('Obraz szary powiększony')

figure(26)

subplot(3,1,1)

imshow(IS12)

title('Obraz rgb pomniejszony')

subplot(3,1,2)

imshow(IS22)

title('Obraz rgb pomniejszony')

subplot(3,1,3)

imshow(IS32)

title('Obraz rgb pomniejszony')

****

Rys. 26. Obraz szary powiększony

****

Rys. 27. Obraz barwny pomniejszony

Funkcją *imresize* wymiar obrazu szarego został powiększony o 1,5, a wymiar obrazu barwnego został pomniejszony o 0,5. Właściwości modeli interpolacyjnych są takie same jak w podpunkcie a).

1. **reshape (tylko dla obrazu w odcieniach szarości, zmodyfikować rozmiar macierzy)**

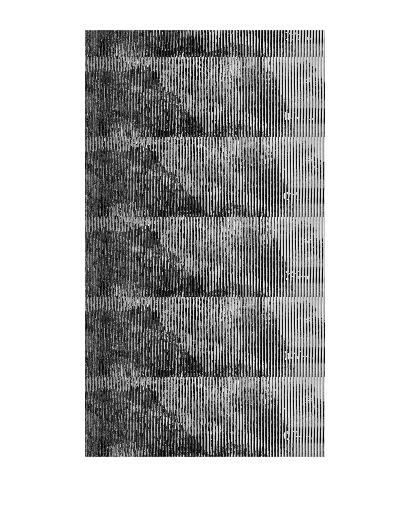
%% Reshape

[i,j]=size(o\_gray);

Br=reshape(o\_gray,j,i);

figure(27)

imshow(Br)



Rys. 28. Modyfikacja rozmiaru obrazu przy pomocy funkcji reshape

B = reshape(A,m,n) zwraca macierz B mxn. Liczba pikseli jest taka sama jak w obrazie wejściowym z tą różnicą, że liczba kolumn i wierszy jest inna. W powyższym przykładzie liczba pikseli w pionie została zamieniona z liczbą pikseli w poziomie.

1. **przetestować działanie funkcji flipud, fliplr,**

%% Funkcje flipud, fliplr

Be=flipud(o\_gray);

Be2=fliplr(o\_gray);

figure(28)

subplot(1,2,1)

imshow(Be);

title('zamiana kolumn')

subplot(1,2,2)

imshow(Be2);

title('zamiana wierszy')

****

Rys. 29. Przykład użycia funkcji obrotu

Funkcja *flipud* służy do zamiany kolejności wartości kolumn od góry do dołu, a funkcja *fliplr* służy do zmiany kolejności wartości wierszy od lewej do prawej. W rezultacie w pierwszym przypadku otrzymano obraz odwrócony do góry nogami, a w drugim przypadku lustrzane odbicie.

1. **imcrop (obraz oryginalny podzielić na dowolną część bloków, następnie za pomocą funkcji cat połączyć ponownie w jeden obraz)**

%% Funkcja imcrop

[w,k]=size(o\_gray); % wymiary obrazu w odcieniach szarości

a=k/5; % obraz dzielony na 5 części równe

figure(29)

title('Obraz sary podzielony na 5 części')

for ii = 1:5

subplot(1,5,ii);

cropp=imcrop(o\_gray,[(ii-1)\*a 0 a w]);

imshow(cropp); % druga część obrazu

title(ii)

end

k1=imcrop(o\_gray,[0 0 a w]);

k2=imcrop(o\_gray,[a 0 a w]);

k3=imcrop(o\_gray,[2\*a 0 a w]);

k4=imcrop(o\_gray,[3\*a 0 a w]);

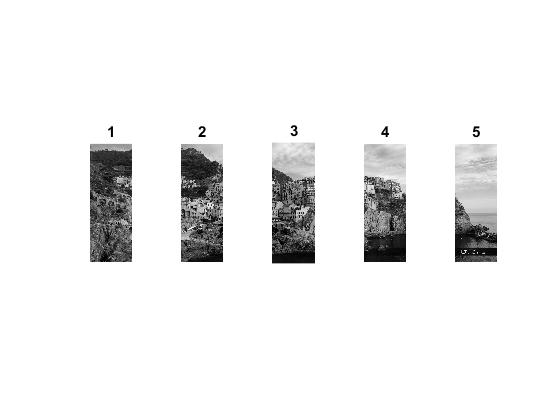
k5=imcrop(o\_gray,[4\*a 0 a w]);

%%scalony

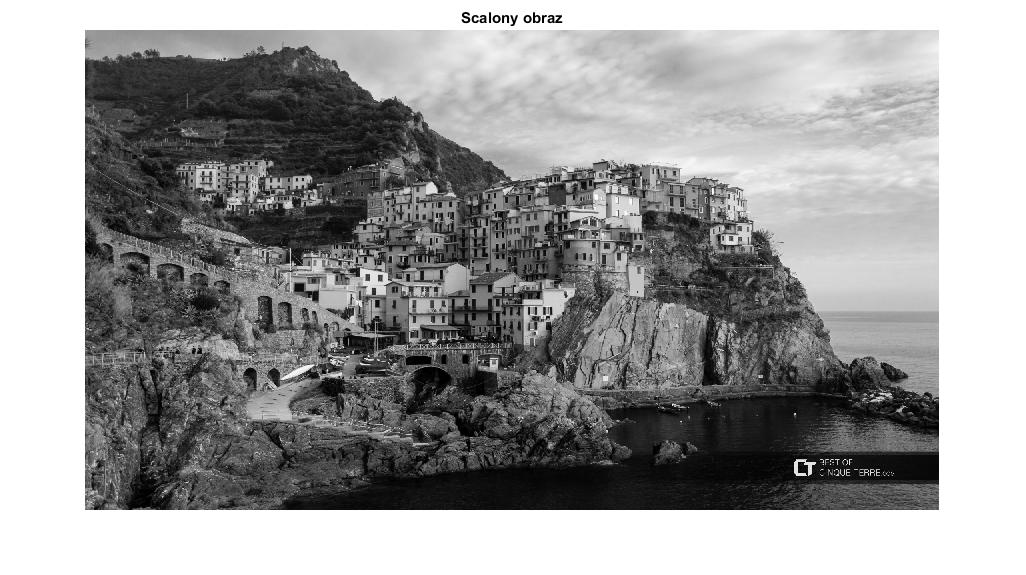
k=cat(2,k1,k2,k3,k4,k5);

figure(30)

imshow(k)

title('Scalony obraz')

Rys. 30. Obraz w skali szarości podzielony na 5 części



Rys. 31. Obraz scalony z 5 kawałków

Obraz został podzielony funkcją *crop* w pętli for na 5 równych kawałków w pionie. Następnie otrzymane kawałki ponownie scalono w obraz bazowy w pętli for.

**6. Dla dowolnych obrazów przeprowadzić operacje (zarówno dla obrazów odcieniach szarości jak i obrazów barwnych):**

**a) suma dwóch obrazów (funkcja** **imadd)**

%% Operacje arytmetyczne

%Suma

I1 = imread('czolg.bmp');

J1 = imread('miasto.jpg');

I2 = rgb2gray(I1);

J2= rgb2gray(J1);

K1 = imadd(I1,J1);

K2 = imadd(I2,J2);

figure(31)

subplot(1,2,1)

imshow(K1)

title('suma obrazów barwnych')

subplot(1,2,2)

imshow(K2)

title('suma obrazów szarych')



Rys. 32. Suma dwóch obrazów

Z wykresu widać, że po operacji sumy obrazy rozjaśniły się i można z nich odczytać zarys poszczególnych elementów składowych.

**b) różnica dwóch obrazów (funkcja** **imabsdiff)**

%% Różnica

K3 = imabsdiff(I1,J1);

K4 = imabsdiff(I2,J2);

figure(32)

subplot(1,2,1)

imshow(K3)

title('różnica obrazów barwnych')

subplot(1,2,2)

imshow(K4)

title('różnica obrazów szarych')

****

Rys. 33. Różnica dwóch obrazów

Po operacji różnicy obraz wynikowy stał się zniekształcony i ciężko jest z niego cokolwiek odczytać. Zmieniła się paleta barw,, a kolory stały się ciemniejsze.

**c) średnia arytmetyczna dwóch obrazów (imdivide)**

%% średnia arytmetyczna

K5 = imdivide(I1,J1);

K6 = imdivide(I2,J2);

figure(33)

subplot(1,2,1)

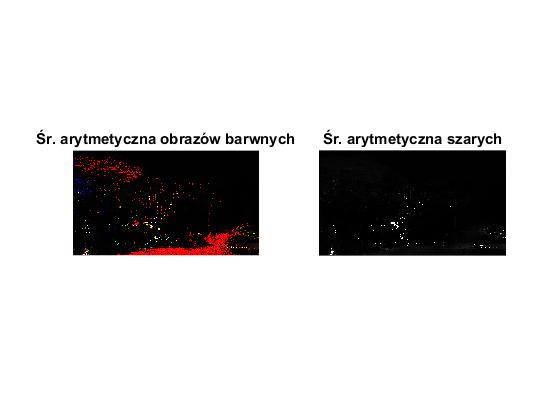
imshow(K5)

title('Śr. arytmetyczna obrazów barwnych')

subplot(1,2,2)

imshow(K6)

title('Śr. arytmetyczna obrazów szarych')



Rys. 34. Średnia arytmetyczna dwóch obrazów

Po operacji średniej arytmetycznej na obrazach dominuje czerń. Na obrazie barwnym widać niewielkie plamy barwy czerwonej i niebieskiej, a na obrazie szarym kilka białych punktów.

**e) iloczyn dwóch obrazów (immultiply)**

%% Iloczyn

K7 = immultiply(I1,J1);

K8 = immultiply(I2,J2);

figure(34)

subplot(1,2,1)

imshow(K7)

title('Iloczyn obrazów barwnych')

subplot(1,2,2)

imshow(K8)

title('Iloczyn szarych')



Rys. 35. Iloczyn dwóch obrazów

Iloczyn dwóch obrazów wygląda odwrotnie niż średnia arytmetyczna. Obszary czarne zostały zastąpione bielą, a obszary czerwone obszarami błękitnymi.

# Zadanie 2

**Na podstawie inicjałów należy ustalić indywidualne parametry do niektórych zadań (Nazwisko Imię – Bączyk Jakub - litera B - pozycja w polskim alfabecie nr 3 (N=3), litera J - pozycja w polskim alfabecie nr 13 (I=13).**

**1. Wczytać dowolny obraz (RGB) a następnie przekształcić go do postaci obrazu binarnego:**

**a) BW = im2bw(I, level)**

**b) BW = im2bw(X, map, level)**

**c) BW = im2bw(RGB, level)**

level- poziom/próg binaryzacji określić na podstawie histogramu obrazu w odcieniach szarości - nasycenie/odcień szarości o największej liczbie pikseli (ponadto należy opisać jakie informacje są zawarte na histogramie)

%% Zadanie2

%% Przeksztalcenie do postaci binarnej

obrazRGB=imread('lew.png');

[Y,map]=imread('lew.png');

obrazG=rgb2gray(obrazRGB);

I=13;

N=3;

%prog przyporzadkowania

H\_I=imhist(obrazG);%historgram

[max\_I,ind]=max(H\_I);

level=(ind)/255;

BW\_a=im2bw(obrazG,level); %obraz intensywnosci do bin

BW\_b=im2bw(Y,map,level); %obraz indexowany do bin

BW\_c=im2bw(obrazRGB,level); %rgb do bin

figure(35)

subplot(2,2,1);

imshow(obrazG);

title('Obraz szary');

subplot(2,2,2);

imshow(BW\_a);

title('BW=im2b2(I,level)');

subplot(2,2,3);

imshow(BW\_b);

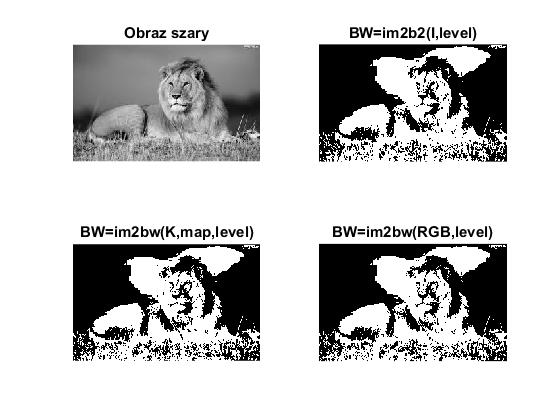
title('BW=im2bw(K,map,level)');

subplot(2,2,4);

imshow(BW\_c);

title('BW=im2bw(RGB,level)');

W celu binaryzacji obraz barwny został przekształcony do obrazu w skali szarości. Następnie na podstawie histogramu i wybraniu nasycenia o największym stopniu szarości wyznaczono poziom binaryzacji. Wyniósł on 0.5373. Obiekt określony jest kolorem biały i ma wartość 1, a tło kolorem czarnym i ma wartość 0.



Rys. 36. Przekształcenie obrazu do postaci binarnej

**d) dla obrazu w odcieniach szarości wykorzystać funkcję graythresh (opisać w jaki sposób określana jest wartość progu - metodę Otsu)**

Obraz kolorowy jest zamieniany na szary, które zapisywany jest na 8 bitach(0-255). Wyznaczamy następnie pewien próg i wartości powyżej tego progu zamieniane są na 1, a poniżej progu na 0. W zależności od progu wyświetlane są piksele białe i czarne.

%% Graythresh

obrazRGB=imread('lew.png');

[Y,map]=imread('lew.png');

obrazG=rgb2gray(obrazRGB);

level2=graythresh(obrazG);

BW\_GRAY=im2bw(obrazG,level2);

figure(36)

subplot(1,2,1);

imshow(obrazG);

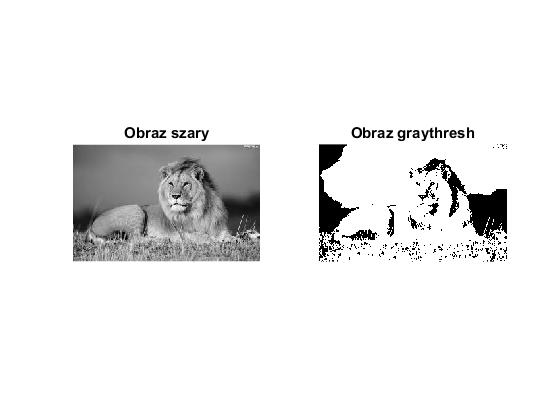
title('Obraz szary');

subplot(1,2,2);

imshow(BW\_GRAY);

title('Obraz graythresh');

Metoda Otsu zmniejsza wariancję dwóch powstałych na histogramie klas. Binaryzacja metodą Otsu przyniosła gorsze rezultaty niż odczytanie progu binaryzacji z histogramu. W tym przypadku próg wyniósł 0.4235 i zbyt duża część tła została potraktowana jako obiekt.



Rys. 37. Binaryzacja obrazu z wykorzystaniem metody progowania Otsu

**e) dla obrazu w odcieniach szarości dokonać binaryzacji od wartości nasycenia równej 100 do 200 z krokiem dla I>N =20, w przeciwnym wypadku krok ustali na wartość 25 (binaryzacja z dolnym progiem)**

%% Binaryzacja

j=1;

for i=110:20:210

BW\_e(:,:,:,j)=im2bw(obrazG,i/255); %wiersze,kolumny,głębia obrazu,j-numer binaryzowanego obrazu

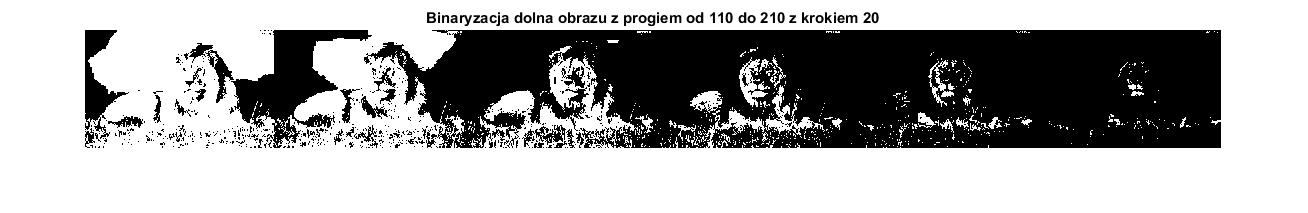
j=j+1;

end

figure(37)

montage(BW\_e,'Size',[1 6]);

title('Binaryzacja dolna obrazu z progiem od 110 do 210 z krokiem 20');



Rys. 38. Binaryzacja dolna obrazu z progiem wartości nasycenia od 110 do 210 z krokiem 20

Im większy jest próg binaryzacji, tym więcej czarnych pikseli zawiera obraz. Najlepiej binaryzacja wypadła dla nasycenia o wartości 150.

**2. Dowolny obraz w odcieniach szarości (szary) podzielić na:**

**a) 4 równe części (fragmenty-bloki)**

%% Dzielenie na 4 czesci

[w k q] =size(obrazG);

p1=obrazG(1:w/2,1:k/2);

p3=obrazG(w/2+1:w,1:k/2);

p2=obrazG(1:w/2,k/2+1:k);

p4=obrazG(w/2+1:w,k/2+1:k);

figure(38)

subplot(2,2,1)

imshow(p1);

subplot(2,2,2);

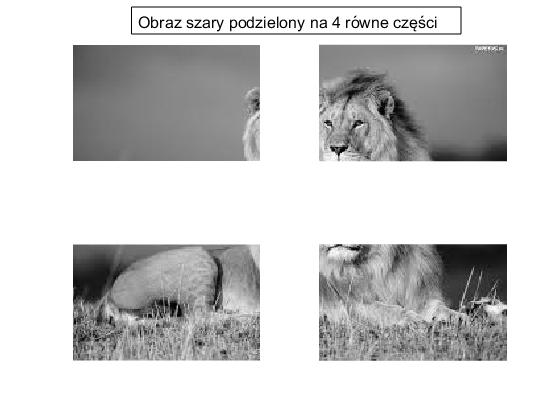
imshow(p2);

subplot(2,2,3);

imshow(p3);

subplot(2,2,4);

imshow(p4);



Rys. 39. Obraz szary podzielony na 4 równe części

**b) (I+N) równych części (fragmenty-bloki), następnie dla każdego z bloków dokonać wyrównywania histogramu (funkcja histeq). Przedstawić i opisać różnice w odniesieniu do oryginalnego obrazu i jego histogramu oraz obrazu oryginalnego (całego) po operacji**

%% Dzielenie na 16 roznych czesci

I=13;

N=3;

for i=1:(I+N)

p(:,:,i)= obrazG(:,(i-1)\*k/(I+N)+1:(i)\*k/(I+N));

p\_h(:,:,i)= histeq(p(:,:,i));

%Obraz podzielony na I+N równych części

figure(39)

subplot(1, (I+N),i)

imshow(p(:,:,i));

% podzielony na równe części z kazdym oddzielnym wyrównaniem

figure(40)

subplot(1,(I+N), i)

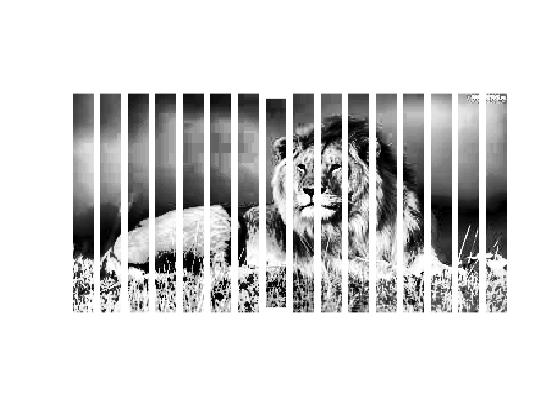
imshow(p\_h(:,:,i));

end

Obraz został podzielony na 16 równych pionowych części. Na skutek wyrównania histogramu dla każdego bloku, każdy blok reprezentuje ten sam poziom intensywności szarości. Na obrazie po modyfikacji obiekt stał się bardziej wyraźny i bardziej kontrastuje z tłem niż na obrazie oryginalnym.



Rys. 40. Obraz oryginalny w skali szarości podzielony na 16 części



Rys. 41. Obraz wynikowy w skali szarości podzielony na 16 części

**3. Operacje morfologiczne. Stwórz obraz w dowolnym programie graficznym na którym są takie elementy jak: zatoczki w obiektach, inicjały (po trzy pierwsze litery imienia i nazwiska), wypustki dla dowolnego obiektu, dwa stykające obiekty, kilka obiektów o różnym kolorze, kształcie i wielkości.**

%% Zadanie 3

R=imread('zad3.png');

R2=rgb2gray(R);

level=graythresh(R2);

R\_bw=im2bw(R,level);

figure(41);

subplot(1,3,1);

imshow(R);

title('Obraz orginalny');

subplot(1,3,2);

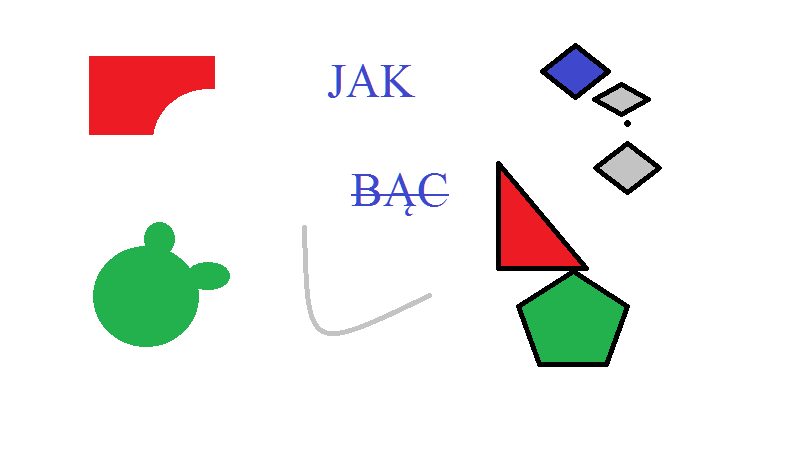
imshow(R2)

title('Obraz w odcieniach szarości');

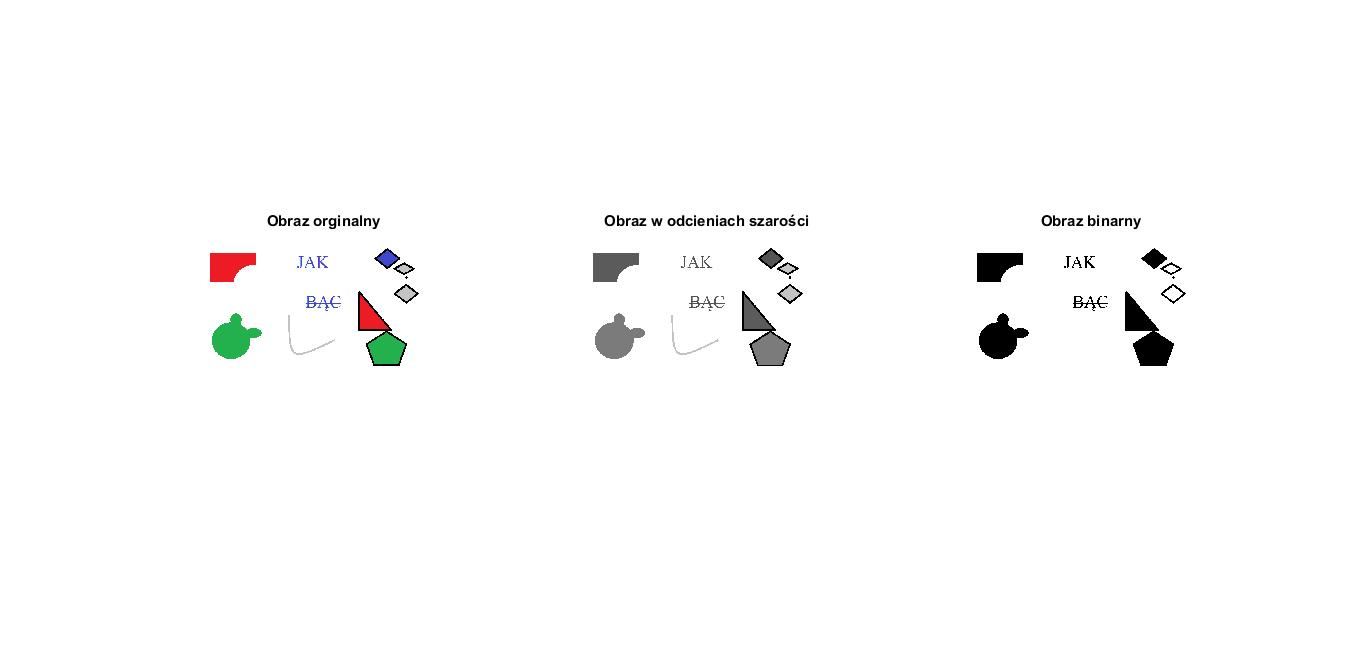
subplot(1,3,3);

imshow(R\_bw);

title('Obraz binarny')



Rys. 42. Obraz oryginalny utworzony w Paincie



Rys. 43. Obraz oryginalny barwny, obraz w skali szarości i obraz binarny

W procesie binaryzacji elementy, które były narysowane jaśniejszym kolorem stały się białe. Kontury które były zaznaczone na czarno, zostały przeniesiono na obraz binarny. Tekst oznaczony kolorem niebieskim oraz figury bez konturów w kolorach czerwonym i zielonym również zostały przeniesione na obraz binarny.

**a) Za pomocą funkcji 'strel' wygenerować wszystkie możliwe elementy strukturalne. Opisać je skrótowo + obrazy i postać macierzowa**

%% Strel

SE1=strel('diamond',5);

SE2=strel('disk',10,6);

SE3=strel('line',30,25);

SE4=strel('octagon',9);

SE5=strel('rectangle',[15,20]);

SE6=strel('square',15);

SE7=strel('arbitrary',[0,1,1,1,0,1]);

SE8=strel('pair',[2,-1]);

SE9=strel('cube',15);

%

figure(42)

subplot(3,3,1);

imshow(getnhood(SE1),'InitialMagnification','fit');

title('Diamond');

subplot(3,3,2);

imshow(getnhood(SE2),'InitialMagnification','fit');

title('Disc');

subplot(3,3,3);

imshow(getnhood(SE3),'InitialMagnification','fit');

title('Line');

subplot(3,3,4);

imshow(getnhood(SE4),'InitialMagnification','fit');

title('Octagon');

subplot(3,3,5);

imshow(getnhood(SE5),'InitialMagnification','fit');

title('Rectangle');

subplot(3,3,6);

imshow(getnhood(SE6),'InitialMagnification','fit');

title('Square');

subplot(3,3,7);

imshow(getnhood(SE7),'InitialMagnification','fit');

title('Arbitrary');

subplot(3,3,8);

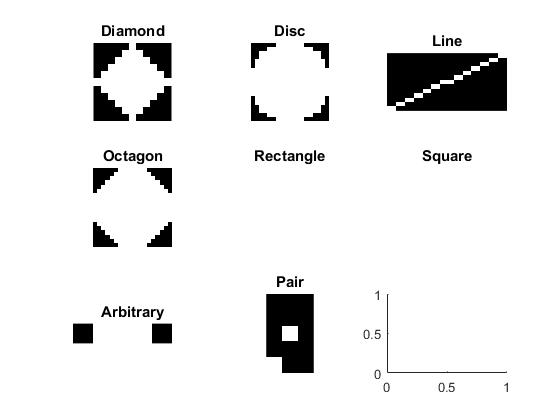
imshow(getnhood(SE8),'InitialMagnification','fit');

title('Pair');

subplot(3,3,9);

imshow(getnhood(SE9),'InitialMagnification','fit');

title('Cube');



Rys. 43. Wygenerowanie elementów strukturalnych

**b) Dla uzyskanego obrazu dokonać operacji: erozji, dylatacji, zamknięcia, otwarcia z uwzględnieniem doboru elementu strukturalnego (pokazać na rysunku, opisać, uzasadnić)**

%% b) operacja erozji

A\_er2=imerode(R\_bw,SE1);

figure(43)

imshow(A\_er2);

title('Operacja erozji')

%operacja dylatacji - odwrotność erozji

A\_odw2=imdilate(R\_bw,SE1);

figure(44)

imshow(A\_odw2);

title('Operacja dylatacji')

A\_otw2=imopen(R\_bw,SE1);

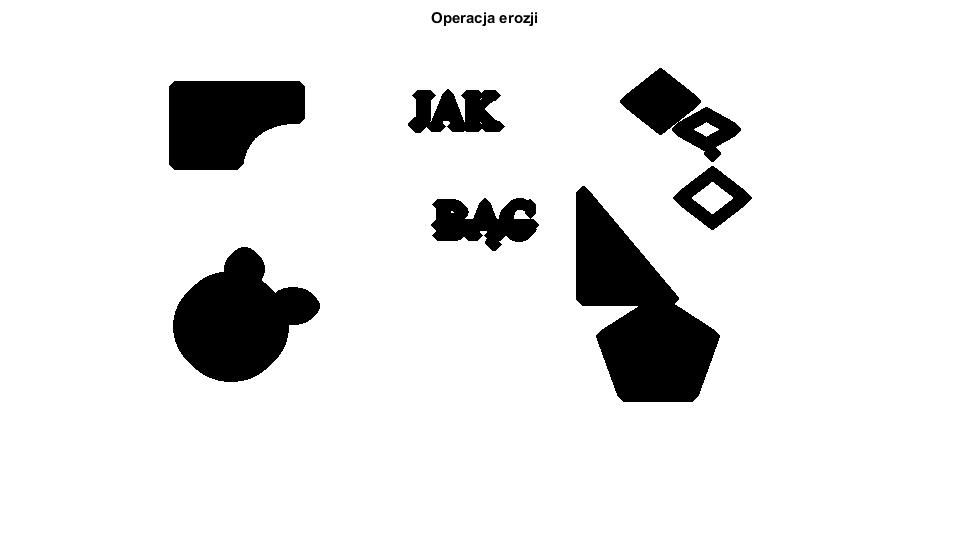
figure(45)

imshow(A\_otw2);title('Operacja otwarcia')

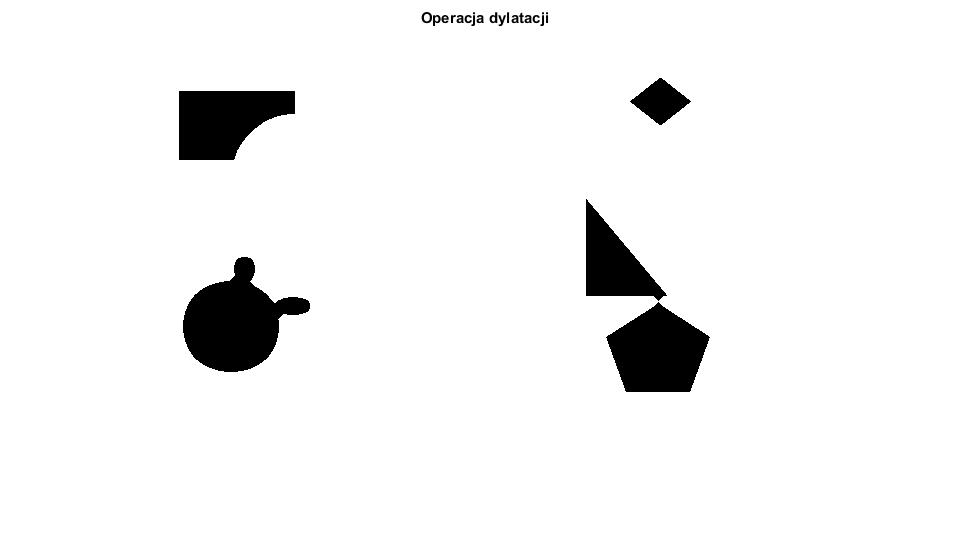
A\_zam2=imclose(R\_bw,SE1);

figure(46)

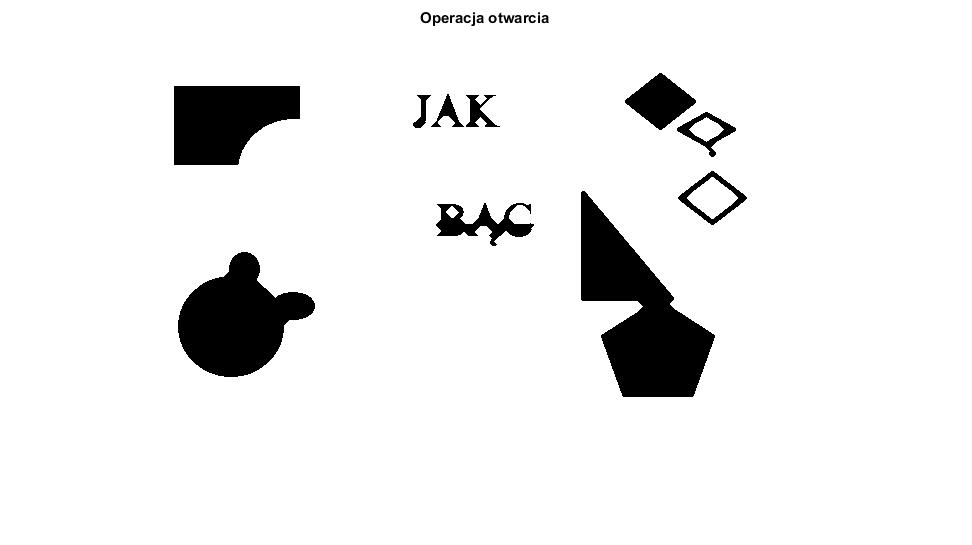
imshow(A\_zam2);title('Operacja zamknięcia')



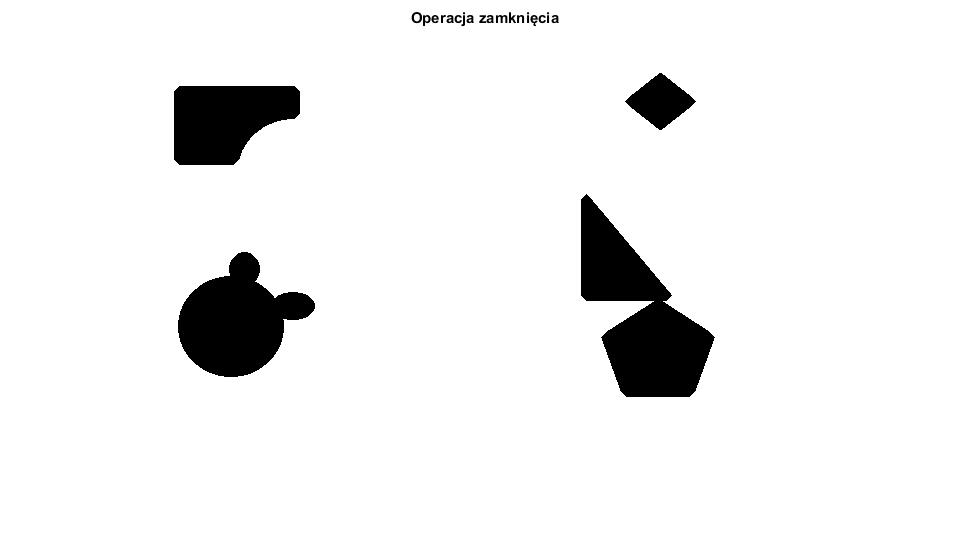
Rys. 44. Operacja erozji z wykorzystaniem elementu strukturalnego diamond



Rys. 45. Operacja dylatacji z wykorzystaniem elementu strukturalnego diamond



Rys. 46. Operacja otwarcia z wykorzystaniem elementu strukturalnego diamond



Rys. 47. Operacja zamknięcia z wykorzystaniem elementu strukturalnego diamond

Operacje erozji, dylatacji, otwarcia i zamknięcia dokonano z użyciem elementu strukturalnego *diamond*, ponieważ dawał on najlepsze rezultaty.Operacja erozji powoduje pogrubienie konturów obiektów na obrazie. Elementy położone blisko siebie zlały się ze sobą, pogrubiła się również czcionka. Operacja dylatacji jest operacją odwrotną do erozji, kontury są pomniejszane. Elementy które posiadały tylko kontury, a nie były wypełnione oraz czcionka zniknęły. Otwarcie to wykonanie erozji a następnie dylatacji. Operacja zamknięcia działa w odwrotnej kolejności.

**4. Stworzyć film (sekwencja klatek) spełniający poniższe kryteria:**

**- zasymulować ruch dwóch różnych pod względem kształtu obiektów po dowolnej trajektorii, w którym w dowolnym momencie dochodzi do styku (zachodzenia) dwóch obiektów,**

**- ruch niejednostajny (obiekty raz poruszają się szybciej, raz zwalniają),**

**- w trakcie przemieszczania dochodzi do zmiany barwy oraz wielkości powierzchni jednego z obiektów**

**Wybrane wyniki przedstawić w postaci złożenia klatek funkcja (montage)):**

**a) dla określonej klatki (numer klatki = Imię - pozycja z alfabetu) wydobyć wszystkie możliwe informacje uzyskiwane za pomocą komendy regionprops (dokładnie opisać znaczenie tych danych-parametrów - jaką informację przenoszą, jeśli istnieje taka możliwość wyświetlić na oddzielnych obrazach z obliczonym-wyznaczonym naniesionym na obraz parametrem);**

%% Zadanie 4

v=VideoReader ('Film2.avi')

A=read(v,3);

info=aviinfo('Film2.avi')

filmG=rgb2gray(A);

level=graythresh(filmG); %Ustalenie progu met Otsu->optymalny próg binaryzacji

A\_bw=~im2bw(filmG,level); %binaryzacja z progiem z metody Otsu

figure(47)

subplot(3,1,1);

imshow(A);

title('3 klatka filmu-obraz RGB')

subplot(3,1,2);

imshow(filmG);

title('3 klatka filmu-w odcieniach szarości')

subplot(3,1,3);

imshow(A\_bw);

title('3 klatka filmu po binaryzacji i negacji');

figure(48)

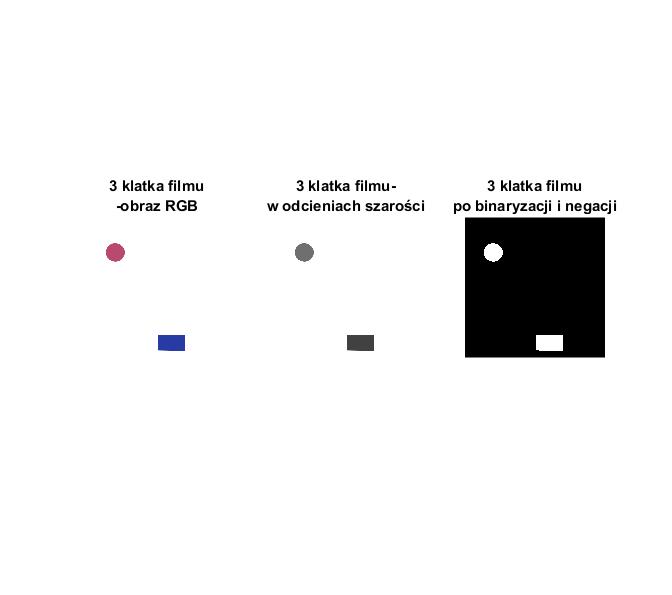
imshow(A\_bw);

s=regionprops(A\_bw,'Centroid');

centroids=cat(1,s.Centroid);

hold on;title('Środek ciężkości obektu');

plot(centroids(:,1),centroids(:,2),'b+','Markersize',15);



Rys. 48. 3 klatka filmu jako obraz RGB, w odcieniu szarości i po binaryzacji i negacji

info=

Filename: 'C:\Users\lenovo\Desktop\Projekt obrazy 2\Film2.avi'

FileSize: 30003200

FileModDate: '01-cze-2017 15:11:25'

NumFrames: 40

FramesPerSecond: 12.0482

Width: 500

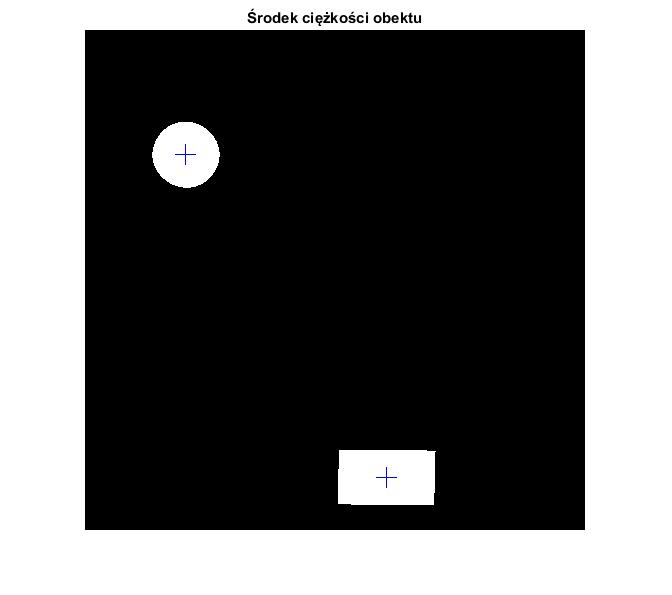
Height: 500

ImageType: 'truecolor'

VideoCompression: 'none'

Quality: 0

NumColormapEntries: 0



Rys. 49. Obiekty wraz ze środkami ciężkości

Do prezentacji wybrano 3 klatkę filmu. Aby zbinaryzować obraz i wyświetlić na nim obiekty, należało zanegować wartości na obrazie. Funkcja aviinfo wyświetla informację o lokalizacji pliku, jego rozdzielczości, dacie modyfikacji, liczbie klatek, liczbie klatek na sekundę, typie kolorów, rozmiarze, kompresji i jakości. Na obrazie wykryto 2 obiekty oraz zaznaczono ich środki ciężkości.

**b) dla klatki o numerze 3 wykorzystać funkcję "bwlabel" - na podstawie tej komendy określić liczbę obiektów, wyjaśnić składnię tej komendy i znaczenie tej funkcji**

%%

[A\_m,liczba]=bwlabel(A\_bw);

figure(49)

imshow(A);

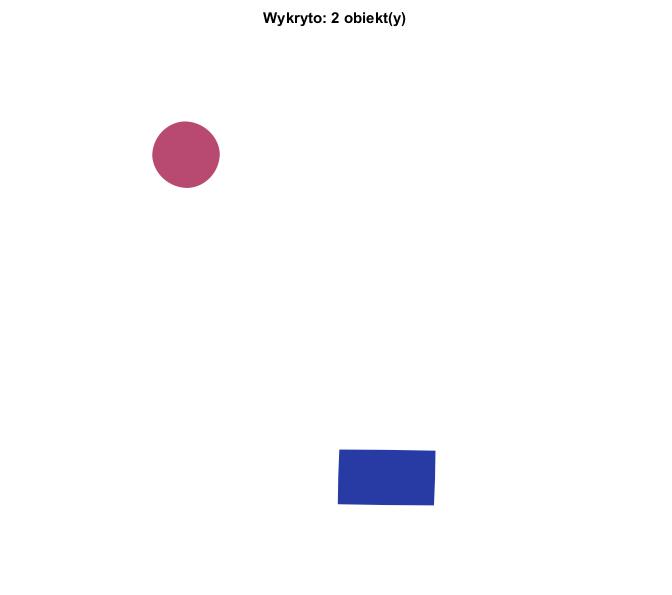
title(['Wykryto: ' num2str(liczba) ' obiekt(y)']);

Pole=bwarea(A\_bw)%pole w pikselach/porównań z regionprops

Pole =

8.7555e+03

Na klatce nr 3 wykryto 2 obiekty. Ich łączne pole wynosi 8.7555e+03 pikseli.



Rys. 50. Wykrycie 2 obiektów na klatce 3

**c) wyznaczyć tor ruchu każdego z obiektów**

for i=1:1:40

A1=read(v,i);

A1\_gg=rgb2gray(A1);

level1=graythresh(A1\_gg);

A1\_bw=~im2bw(A1\_gg,level1);

c1=regionprops(A1\_bw,'Centroid')

c12=struct2cell(c1);

c11=cell2mat(c12);

figure(50);

plot(-c11(1,1),c11(1,3),'k+','Markersize',15)

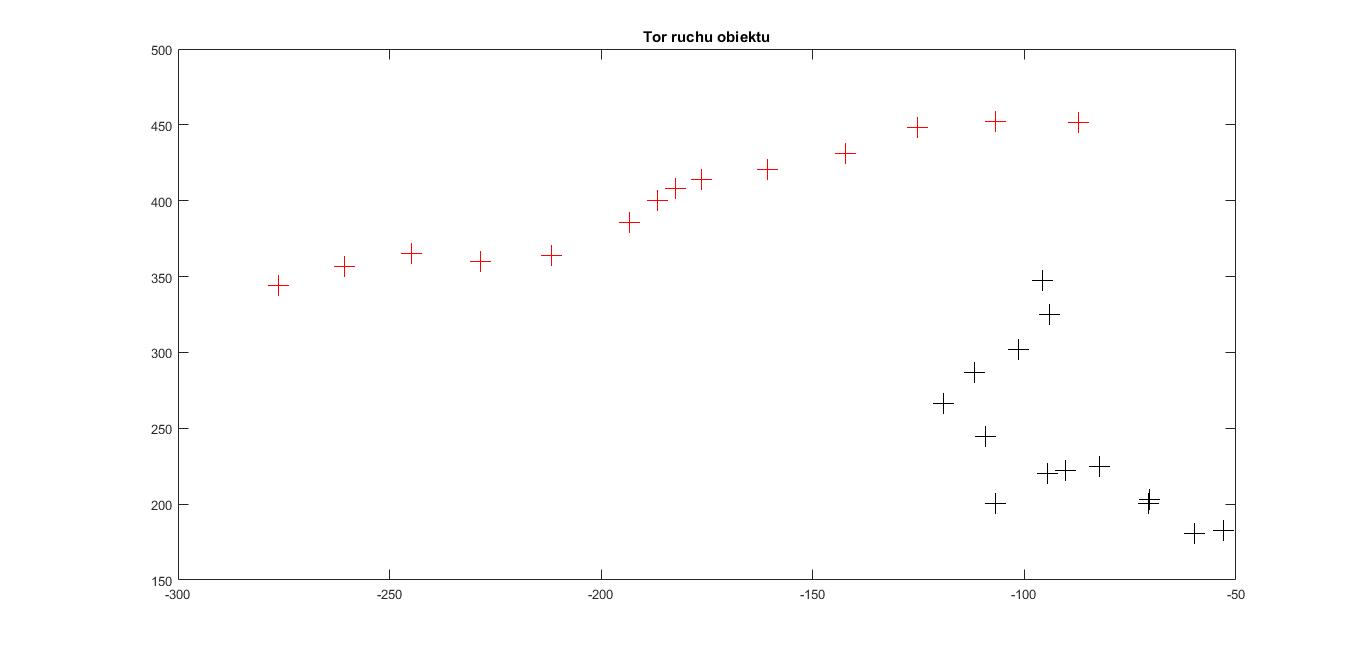
hold on

plot(-c11(1,2),c11(1,4),'r+','Markersize',15);

hold on

title('Tor ruchu obiektu');

end



Rys. 51. Wyznaczenie toru ruchu obiektów

Po wyznaczeniu środków ciężkości przedstawiono je na wykresie w pętli jako tor przemieszczenia się obiektów.

**Wnioski**

W zadaniu pierwszym na początku wczytano obrazy o różnych rozszerzeniach. Następnie omówiono funkcje do wyświetlania obrazów w Matlabie oraz zapisano obrazy ze zmienionymi rozszerzeniami. Przedstawiono obraz dla różnych modeli barwnych oraz dokonano na nim przekształceń takich jak pierwiastkowanie, potęgowanie i mnożenie przez stałą. Obraz konwertowano z formatu rgb na monochromatyczny, binarny i indeksowany.

Dla obrazów barwnych wyznaczono histogramy przedstawiające zależność ilości pikseli od nasycenia oraz reprezentacje przestrzenne nasycenia poszczególnych składowych RGB. W tym celu konieczna była zmiana rozmiaru obrazu, ponieważ początkowo jego rozdzielczość była dosyć duża, co znacznie wydłużało obliczenia. Podzielenia wartości kanału niebieskiego na obrazie spowodowało zmianę barwy obrazu na zielony. Po zastosowaniu funkcji pseudokolorowania obrazy stały się zniekształcone i nieczytelne. Zastosowano trzy palety barw o trzech stopniach ilości kolorów.

Następnie wykonano obroty obrazów o zadany kąt przy zastosowaniu różnych funkcji interpolacyjnych. Zastosowano takie operacje jak zmiana rozmiaru obrazu wynikowego, obrócenie obrazu, lustrzane odbicie czy podział obrazu na kawałki, a następnie połączenie w całość. Na koniec przeprowadzono operacje sumy, różnicy, dzielenia oraz mnożenia dwóch obrazów.

W zadaniu drugim skupiono się głównie na binaryzacji obrazu. Binaryzacja obrazu pozwala na dokładną analizę danych zawartych na obrazie poprzez oddzielenie elementów od tła. Pozwala to na zbieranie informacji o położeniu, wielkości, ilości czy środku ciężkości detali obrazu. Przy binaryzacji należy pamiętać o dobraniu właściwego progu aby elementy np. jasne nie były traktowane jako część tła. Odpowiedni prób binaryzacji można wyznaczyć z analizy histogramu lub stosując jedną z dostępnych metod np. metodę Otsu. W programie Matlab należy zastosować negację, ponieważ obiekty mają kolor biały i przyjmują wartość 1. Operacje morfologiczne pozwalają na edycję konturów, wypełnień czy ubytków w obrazie co pozwala na odpowiednie przygotowanie obrazu do prawidłowej obróbki. Do operacji morfologicznych wybrano element strukturalny diamond, ponieważ dawał on najlepsze efekty podczas operacji.

Ostatnim krokiem była praca z filmem. Nagrany w programie Corel RAVE 40-klatkowy film został zbinaryzowany. Umożliwiło to wychwycenie ilości obiektów w poszczególnych klatkach, wyświetlenie ich pola, czy też środków ciężkości. Na podstawie położenia środków ciężkości można było wyznaczyć tor przemieszczenia się obiektów.

Oprogramowanie Matlab posiada wiele wbudowanych funkcji do analizy i przetwarzania obrazów. Jest to bardzo rozbudowane narzędzie, które ułatwia pracę z obrazami. Do najtrudniejszych zadań należał dobór właściwych parametrów funkcji, jak również pisanie podprogramów do wykonywania wielu operacji w jednym czasie. Projekt nie należał do najtrudniejszych, jednak ilość poleceń, opis i zrozumienie wszystkich funkcji oraz często sprawdzenie ich wszystkich parametrów okazało się bardzo czasochłonne i wymagało dużego nakładu pracy.

clc;

clear all;

close all;

%% Wczytanie obrazów

obraz\_jpg=imread('miasto.jpg');

obraz\_bmp=imread('czolg.bmp');

obraz\_tiff=imread('napis.tiff');

obraz\_png=imread('lew.png');

[A, map]=imread('pingwin.gif',1:15);

%% Wyświetlenie informacji o obrazach

info\_jpg=imfinfo('miasto.jpg')

info\_bmp=imfinfo('czolg.bmp');

info\_tif=imfinfo('napis.tiff');

info\_png=imfinfo('lew.png');

imfinfo('pingwin.gif');

info=imfinfo('miasto.jpg')

if info.Height < 400

afterInfo = imrotate(obraz\_jpg, 90);

else

afterInfo = imrotate(obraz\_jpg, 180);

end

figure (1)

imshow(afterInfo)

title('Obrót obrazu jpg')

%% Wyświetlenie obrazów

figure(1)

subplot(2,3,1)

imshow(obraz\_jpg);

title('Obraz JPEG')

subplot(2,3,2)

imshow(obraz\_bmp);

title('Obraz BMP')

subplot(2,3,3)

imshow(obraz\_tiff);

title('Obraz TIFF')

subplot(2,3,4)

imshow(obraz\_png);

title('Obraz PNG')

subplot(2,3,5)

for i=1:8 %wyświetlenie 8 klatek GIF

imshow(A(:,:,:,i),map)

end

title('Obraz GIF');

%% Porównanie funkcji

figure(2)

subplot(2,2,1)

imshow(obraz\_jpg);

title('Funkcja imshow')

subplot(2,2,2)

image(obraz\_jpg);

title('Funkcja image')

subplot(2,2,3)

imagesc(obraz\_jpg);

title('Funkcja imagesc')

subplot(2,2,4)

warp(obraz\_jpg);

title('Funkcja warp')

%% Zapisanie JPG w innych rozszerzeniach

imwrite(obraz\_jpg,'zJPGdoBMP.bmp'); %zapis do bmp

imwrite(obraz\_jpg,'zJPGdoTIF.tiff'); %zapis do tiff

imwrite(obraz\_jpg,'zJPGdoPNG.png'); %zapis do png

%% Nieliniowe przekształcenie obrazu

%%Pierwiastkowanie

o\_rgb = obraz\_jpg;

o\_drgb = double(o\_rgb)/255;

o\_gray = rgb2gray(obraz\_jpg);

o\_dgray = double(o\_gray)/255;

figure(3)

subplot(2,2,1)

imshow(o\_rgb);

title('Obraz barwny')

subplot(2,2,2)

imshow(sqrt(o\_drgb));

title('Obr. barwny spierwiastkowany');

subplot(2,2,3)

imshow(o\_gray);

title('Obraz w odcieniu szarości')

subplot(2,2,4)

imshow(sqrt(o\_dgray));

title('Obr. szary spierwiastkowany');

%% Potęgowanie

figure(4)

subplot(2,2,1)

imshow(o\_rgb);

title('Obraz barwny')

subplot(2,2,2)

imshow(o\_drgb.^2);

title('Obr. barwny potęgowany');

subplot(2,2,3)

imshow(o\_gray);

title('Obraz w odcieniu szarosci')

subplot(2,2,4)

imshow(o\_dgray.^2);

title('Obr. szary potęgowany');

%% Podwojenie

figure(5)

subplot(2,2,1)

imshow(o\_rgb);

title('Obraz barwny')

subplot(2,2,2)

imshow(o\_drgb.\*2);

title('Obr. barwny podwojony');

subplot(2,2,3)

imshow(o\_gray);

title('Obraz w odcnieu szarosci')

subplot(2,2,4)

imshow(o\_dgray.\*2);

title('Obr. szary podwojony');

%% Konwersja obrazów

%%RGB na hsv

rgb\_hsv=rgb2hsv(o\_rgb);

hsv\_rgb=hsv2rgb(rgb\_hsv);

figure(6)

subplot(1,2,1)

imshow(rgb\_hsv);

title('Obraz HSV')

subplot(1,2,2)

imshow(hsv\_rgb);

title('Obraz RGB');

%%RGB na NTSC

rgb\_ntsc=rgb2ntsc(o\_rgb);

ntsc\_rgb=ntsc2rgb(rgb\_ntsc);

figure(7)

subplot(1,2,1)

imshow(rgb\_ntsc);

title('Obraz NTSC')

subplot(1,2,2)

imshow(ntsc\_rgb);

title('Obraz RGB');

%%RGB na YCbCr

rgb\_ycbcr=rgb2ycbcr(o\_rgb);

ycbcr\_rgb=ycbcr2rgb(rgb\_ycbcr);

figure(8)

subplot(1,2,1)

imshow(rgb\_ycbcr);

title('Obraz YCbCr')

subplot(1,2,2)

imshow(ycbcr\_rgb);

title('Obraz RGB');

%% Wyświetlenie kolorów we współrzędnych 3D

%%Barwy

obrazR=obraz\_jpg(:,:,1); %Kanał czerwony

obrazG=obraz\_jpg(:,:,2); %Kanał zielony

obrazB=obraz\_jpg(:,:,3); %Kanał niebieski

[w,k]=size(obrazR);

%Wyswietlanie kanalu czerwonego

r=0:1:255;

rr=r';

z=zeros(256,3);

z(:,1)=rr;

red=z/255;

%Wyswietlanie kanalu zielonego

r2=0:1:255;

rr2=r2';

z2=zeros(256,3);

z2(:,2)=rr2;

green=z2/255;

%Wyswietlanie kanalu niebieskiego

r3=0:1:255;

rr3=r3';

z3=zeros(256,3);

z3(:,3)=rr3;

blue=z3/255;

figure(9)

mesh(1:k,w:-1:1,obrazR,'facecolor','texturemap','edgecolor','none','cdatamapping','direct');

colormap(red);

view(3);

colorbar('vert');

title('Wykres 3D-kanal czerwony');

xlabel('Kolumna');

ylabel('Wiersz');

zlabel('Natężenie');

figure(10)

mesh(1:k,w:-1:1,obrazG,'facecolor','texturemap','edgecolor','none','cdatamapping','direct');

colormap(green);

view(3);

colorbar('vert');

title('Wykres 3D-kanal zielony');

xlabel('Kolumna');

ylabel('Wiersz');

zlabel('Natężenie');

figure(11)

mesh(1:k,w:-1:1,obrazB,'facecolor','texturemap','edgecolor','none','cdatamapping','direct');

colormap(blue);

view(3);

colorbar('vert');

title('Wykres 3D-kanal niebieski');

xlabel('Kolumna');

ylabel('Wiersz');

zlabel('Natężenie');

%% Histogramy

figure(12)

dane\_hist=imhist(obrazR);

colormap(red);

imhist(obrazR);

colorbar('location','southoutside');

grid on;

title('Histogram - kanal czerwony');

figure(13)

dane\_hist=imhist(obrazG);

colormap(green);

imhist(obrazG);

colorbar('location','southoutside');

grid on;

title('Histogram - kanal zielony');

figure(14)

dane\_hist=imhist(obrazB);

colormap(blue);

imhist(obrazB);

colorbar('location','southoutside');

grid on;

title('Histogram - kanal niebieski');

%% Wykreslenie profilu barwnego RGB wzdłuż wybranego odcinka

figure(15)

imshow(obraz\_jpg);

c=improfile(obraz\_jpg,[730 740],[275 325]);

line([730 740],[275 325],'Color','r','Linewidth',5);

figure(16)

plot(c(:,:,1),'r');

hold on;

plot(c(:,:,2),'g');

hold on;

plot(c(:,:,3),'b');

hold on;

title('Profile barwne RGB wzdłuż danej linii');

grid on;

xlabel('Długość odcinka');

ylabel('Intensywnosc pikseli');

%% Modyfikacja wybranego kanału barwnego

ObrazR=obraz\_jpg(:,:,1);

ObrazG=obraz\_jpg(:,:,2);

ObrazB=obraz\_jpg(:,:,3)/5;

obraz\_mod=cat(3,ObrazR,ObrazG,ObrazB);

figure(17)

imshow(obraz\_mod);

title('Zmodyfikowany niebieski kanał barwny');

%% Pseudokolorowanie

[ObrazRGB]=imread('miasto.jpg');

[x1,map1]=rgb2ind(ObrazRGB,16);%16 ilość poziomów danej palety

[x2,map2]=rgb2ind(ObrazRGB,128);%128 ilość poziomów danej palety

[x3,map3]=rgb2ind(ObrazRGB,255);%255 ilość poziomów danej palety

figure(18)

subplot(1,3,1);

imshow(x1,colormap(prism));

title('16 kolorow');

subplot(1,3,2);

imshow(x2,colormap(prism));

title('128 kolorow');

subplot(1,3,3);

imshow(x3,colormap(prism));

title('255 kolorow');

figure(19)

subplot(1,3,1);

imshow(x1,colormap(copper));

title('16 kolorow');

subplot(1,3,2);

imshow(x2,colormap(copper));

title('128 kolorow');

subplot(1,3,3);

imshow(x3,colormap(copper));

title('255 kolorow');

figure(20)

subplot(1,3,1);

imshow(x1,colormap(jet));

title('16 kolorow');

subplot(1,3,2);

imshow(x2,colormap(jet));

title('128 kolorow');

subplot(1,3,3);

imshow(x3,colormap(jet));

title('255 kolorow');

%% Obracanie obrazu

J11=imrotate(o\_rgb,-45,'nearest','crop');

J12=imrotate(o\_gray,-45,'nearest','loose');

J21=imrotate(o\_rgb,-45,'bilinear','crop');

J22=imrotate(o\_gray,-45,'bilinear','loose');

J31=imrotate(o\_rgb,-45,'bicubic','crop');

J32=imrotate(o\_gray,-45,'bicubic','loose');

figure(21)

subplot(1,2,1)

imshow(J11)

title('Obraz rgb obrócony o 45°')

subplot(1,2,2)

imshow(J12)

title('Obraz szary obrócony o 45°')

figure(22)

subplot(1,2,1)

imshow(J21)

title('Obraz rgb obrócony o 45°')

subplot(1,2,2)

imshow(J22)

title('Obraz szary obrócony o 45°')

figure(23)

subplot(1,2,1)

imshow(J31)

title('Obraz rgb obrócony o 45°')

subplot(1,2,2)

imshow(J32)

title('Obraz szary obrócony o 45°')

%% Maketfrom

T = maketform('affine',[.5 0 0; .5 2 0; 0 0 1]);

tformfwd([10 20],T);

T1 = imtransform(o\_gray,T);

figure(24)

imshow(T1)

%% Imersize

IS12=imresize(o\_rgb,0.5,'nearest');

IS22=imresize(o\_rgb,0.5,'bilinear');

IS32=imresize(o\_rgb,0.5,'bicubic');

IS11=imresize(o\_gray,1.5,'nearest');

IS21=imresize(o\_gray,1.5,'bilinear');

IS31=imresize(o\_gray,1.5,'bicubic');

figure(25)

subplot(3,1,1)

imshow(IS11)

title('Obraz szary powiększony')

subplot(3,1,2)

imshow(IS21)

title('Obraz szary powiększony')

subplot(3,1,3)

imshow(IS31)

title('Obraz szary powiększony')

figure(26)

subplot(3,1,1)

imshow(IS12)

title('Obraz rgb pomniejszony')

subplot(3,1,2)

imshow(IS22)

title('Obraz rgb pomniejszony')

subplot(3,1,3)

imshow(IS32)

title('Obraz rgb pomniejszony')

%% Reshape

[i,j]=size(o\_gray);

Br=reshape(o\_gray,j,i);

figure(27)

imshow(Br)

%% Funkcje flipud, fliplr

Be=flipud(o\_gray);

Be2=fliplr(o\_gray);

figure(28)

subplot(1,2,1)

imshow(Be);

title('zamiana kolumn')

subplot(1,2,2)

imshow(Be2);

title('zamiana wierszy')

%% Funkcja imcrop

[w,k]=size(o\_gray); % wymiary obrazu w odcieniach szarości

a=k/5; % obraz dzielony na 5 części równe

figure(29)

title('Obraz sary podzielony na 5 części')

for ii = 1:5

subplot(1,5,ii);

cropp=imcrop(o\_gray,[(ii-1)\*a 0 a w]);

imshow(cropp); % druga część obrazu

title(ii)

end

k1=imcrop(o\_gray,[0 0 a w]);

k2=imcrop(o\_gray,[a 0 a w]);

k3=imcrop(o\_gray,[2\*a 0 a w]);

k4=imcrop(o\_gray,[3\*a 0 a w]);

k5=imcrop(o\_gray,[4\*a 0 a w]);

%%scalony

k=cat(2,k1,k2,k3,k4,k5);

figure(30)

imshow(k)

title('Scalony obraz')

%% Opracje arytmetyczne

%Suma

I1 = imread('czolg.bmp');

J1 = imread('miasto.jpg');

I2 = rgb2gray(I1);

J2= rgb2gray(J1);

K1 = imadd(I1,J1);

K2 = imadd(I2,J2);

figure(31)

subplot(1,2,1)

imshow(K1)

title('suma obrazów barwnych')

subplot(1,2,2)

imshow(K2)

title('suma obrazów szarych')

%% Różnica

K3 = imabsdiff(I1,J1);

K4 = imabsdiff(I2,J2);

figure(32)

subplot(1,2,1)

imshow(K3)

title('różnica obrazów barwnych')

subplot(1,2,2)

imshow(K4)

title('różnica obrazów szarych')

%% średnia arytmetyczna

K5 = imdivide(I1,J1);

K6 = imdivide(I2,J2);

figure(33)

subplot(1,2,1)

imshow(K5)

title('Śr. arytmetyczna obrazów barwnych')

subplot(1,2,2)

imshow(K6)

title('Śr. arytmetyczna obrazów szarych')

%% Iloczyn

K7 = immultiply(I1,J1);

K8 = immultiply(I2,J2);

figure(34)

subplot(1,2,1)

imshow(K7)

title('Iloczyn obrazów barwnych')

subplot(1,2,2)

imshow(K8)

title('Iloczyn szarych')

%% Zadanie2

%% Przeksztalcenie do postaci binarnej

obrazRGB=imread('lew.png');

[Y,map]=imread('lew.png');

obrazG=rgb2gray(obrazRGB);

I=13;

N=3;

%prog przyporzadkowania

H\_I=imhist(obrazG);%historgram

[max\_I,ind]=max(H\_I);

level=(ind)/255;

%a)BW=im2bw(I,level) obraz intensywnosci do bin

BW\_a=im2bw(obrazG,level); %level z przedziału 0-1;

%b)BW=im2bw(K,map,level) obraz indexowany do bin

BW\_b=im2bw(Y,map,level);

%c) Bw=im2bw(RGB,level) rgb do bin

BW\_c=im2bw(obrazRGB,level);

figure(35)

subplot(2,2,1);

imshow(obrazG);

title('Obraz szary');

subplot(2,2,2);

imshow(BW\_a);

title('BW=im2b2(I,level)');

subplot(2,2,3);

imshow(BW\_b);

title('BW=im2bw(K,map,level)');

subplot(2,2,4);

imshow(BW\_c);

title('BW=im2bw(RGB,level)');

%% Graythresh

obrazRGB=imread('lew.png');

[Y,map]=imread('lew.png');

obrazG=rgb2gray(obrazRGB);

level2=graythresh(obrazG);

BW\_GRAY=im2bw(obrazG,level2);

figure(36)

subplot(1,2,1);

imshow(obrazG);

title('Obraz szary');

subplot(1,2,2);

imshow(BW\_GRAY);

title('Obraz graythresh');

%% Binaryzacja

j=1;

for i=110:20:210

BW\_e(:,:,:,j)=im2bw(obrazG,i/255); %wiersze,kolumny,głębia obrazu,j-numer binaryzowanego obrazu

j=j+1;

end

figure(37)

montage(BW\_e,'Size',[1 6]);

title('Binaryzacja dolna obrazu z progiem od 110 do 210 z krokiem 20');

%% Dzielenie na 4 czesci

[w k q] =size(obrazG);

p1=obrazG(1:w/2,1:k/2);

p3=obrazG(w/2+1:w,1:k/2);

p2=obrazG(1:w/2,k/2+1:k);

p4=obrazG(w/2+1:w,k/2+1:k);

figure(38)

subplot(2,2,1)

imshow(p1);

subplot(2,2,2);

imshow(p2);

subplot(2,2,3);

imshow(p3);

subplot(2,2,4);

imshow(p4);

%% Dzielenie na 16 roznych czesci

I=13;

N=3;

for i=1:(I+N)

p(:,:,i)= obrazG(:,(i-1)\*k/(I+N)+1:(i)\*k/(I+N));

p\_h(:,:,i)= histeq(p(:,:,i));

%Obraz podzielony na I+N równych części

figure(39)

subplot(1, (I+N),i)

imshow(p(:,:,i));

% podzielony na równe części z kazdym oddzielnym wyrównaniem

figure(40)

subplot(1,(I+N), i)

imshow(p\_h(:,:,i));

end

%% Zadanie 3

R=imread('zad3.png');

R2=rgb2gray(R);

level=graythresh(R2);

R\_bw=im2bw(R,level);

figure(41);

subplot(1,3,1);

imshow(R);

title('Obraz orginalny');

subplot(1,3,2);

imshow(R2)

title('Obraz w odcieniach szarości');

subplot(1,3,3);

imshow(R\_bw);

title('Obraz binarny')

%% Strel

SE1=strel('diamond',5);

SE2=strel('disk',10,6);

SE3=strel('line',30,25);

SE4=strel('octagon',9);

SE5=strel('rectangle',[15,20]);

SE6=strel('square',15);

SE7=strel('arbitrary',[0,1,1,1,0,1]);

SE8=strel('pair',[2,-1]);

SE9=strel('cube',5);

%

figure(42)

subplot(3,3,1);

imshow(getnhood(SE1),'InitialMagnification','fit');

title('Diamond');

subplot(3,3,2);

imshow(getnhood(SE2),'InitialMagnification','fit');

title('Disc');

subplot(3,3,3);

imshow(getnhood(SE3),'InitialMagnification','fit');

title('Line');

subplot(3,3,4);

imshow(getnhood(SE4),'InitialMagnification','fit');

title('Octagon');

subplot(3,3,5);

imshow(getnhood(SE5),'InitialMagnification','fit');

title('Rectangle');

subplot(3,3,6);

imshow(getnhood(SE6),'InitialMagnification','fit');

title('Square');

subplot(3,3,7);

imshow(getnhood(SE7),'InitialMagnification','fit');

title('Arbitrary');

subplot(3,3,8);

imshow(getnhood(SE8),'InitialMagnification','fit');

title('Pair');

subplot(3,3,9);

imshow(getnhood(SE9),'InitialMagnification','fit');

title('Cube');

%% b) operacja erozji

A\_er2=imerode(R\_bw,SE1);

figure(43)

imshow(A\_er2);

title('Operacja erozji')

%operacja dylatacji - odwrotność erozji

A\_odw2=imdilate(R\_bw,SE1);

figure(44)

imshow(A\_odw2);

title('Operacja dylatacji')

A\_otw2=imopen(R\_bw,SE1);

figure(45)

imshow(A\_otw2);title('Operacja otwarcia')

A\_zam2=imclose(R\_bw,SE1);

figure(46)

imshow(A\_zam2);title('Operacja zamknięcia')

%% Zadanie 4

v=VideoReader ('Film2.avi')

A=read(v,3);

info=aviinfo('Film2.avi')

filmG=rgb2gray(A);

level=graythresh(filmG); %Ustalenie progu met Otsu->optymalny próg binaryzacji

A\_bw=~im2bw(filmG,level); %binaryzacja z progiem z metody Otsu

figure(47)

subplot(1,3,1);

imshow(A);

title({'3 klatka filmu';'-obraz RGB'})

subplot(1,3,2);

imshow(filmG);

title({'3 klatka filmu-';'w odcieniach szarości'})

subplot(1,3,3);

imshow(A\_bw);

title({'3 klatka filmu'; 'po binaryzacji i negacji'});

figure(48)

imshow(A\_bw);

s=regionprops(A\_bw,'Centroid');

centroids=cat(1,s.Centroid);

hold on;title('Środek ciężkości obektu');

plot(centroids(:,1),centroids(:,2),'b+','Markersize',15);

%%

[A\_m,liczba]=bwlabel(A\_bw);

figure(49)

imshow(A);

title(['Wykryto: ' num2str(liczba) ' obiekt(y)']);

Pole=bwarea(A\_bw)%pole w pikselach/porównań z regionprops

%%

for i=1:1:40

A1=read(v,i);

A1\_gg=rgb2gray(A1);

level1=graythresh(A1\_gg);

A1\_bw=~im2bw(A1\_gg,level1);

c1=regionprops(A1\_bw,'Centroid')

c12=struct2cell(c1);

c11=cell2mat(c12);

figure(50);

plot(-c11(1,1),c11(1,3),'k+','Markersize',15)

hold on

plot(-c11(1,2),c11(1,4),'r+','Markersize',15);

hold on

title('Tor ruchu obiektu');

end