

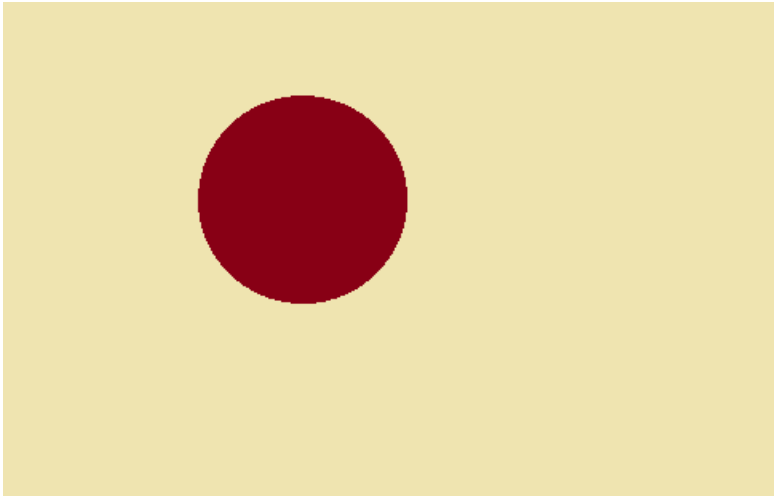
# Sprawozdanie PSIwSUM

## Komputerowe przetwarzanie obrazów cz.2

Urszula Starowicz – 407177

Zadanie 1 – Analiza obrazu: jeden obiekt

Obraz:



Kod:

```
clc;
clear all;
close all;

info = imfinfo('ObrazZ1.png');

A = imread('ObrazZ1.png');
im_size = size(A);
A=rgb2gray(A);

hist = imhist(A);

B = medfilt2(A, [8 8]);

T = 0.5;
C = im2bw(B,T);

D = not(C);

[num, L] = bwlabel(D);
feats = regionprops(D, 'all');

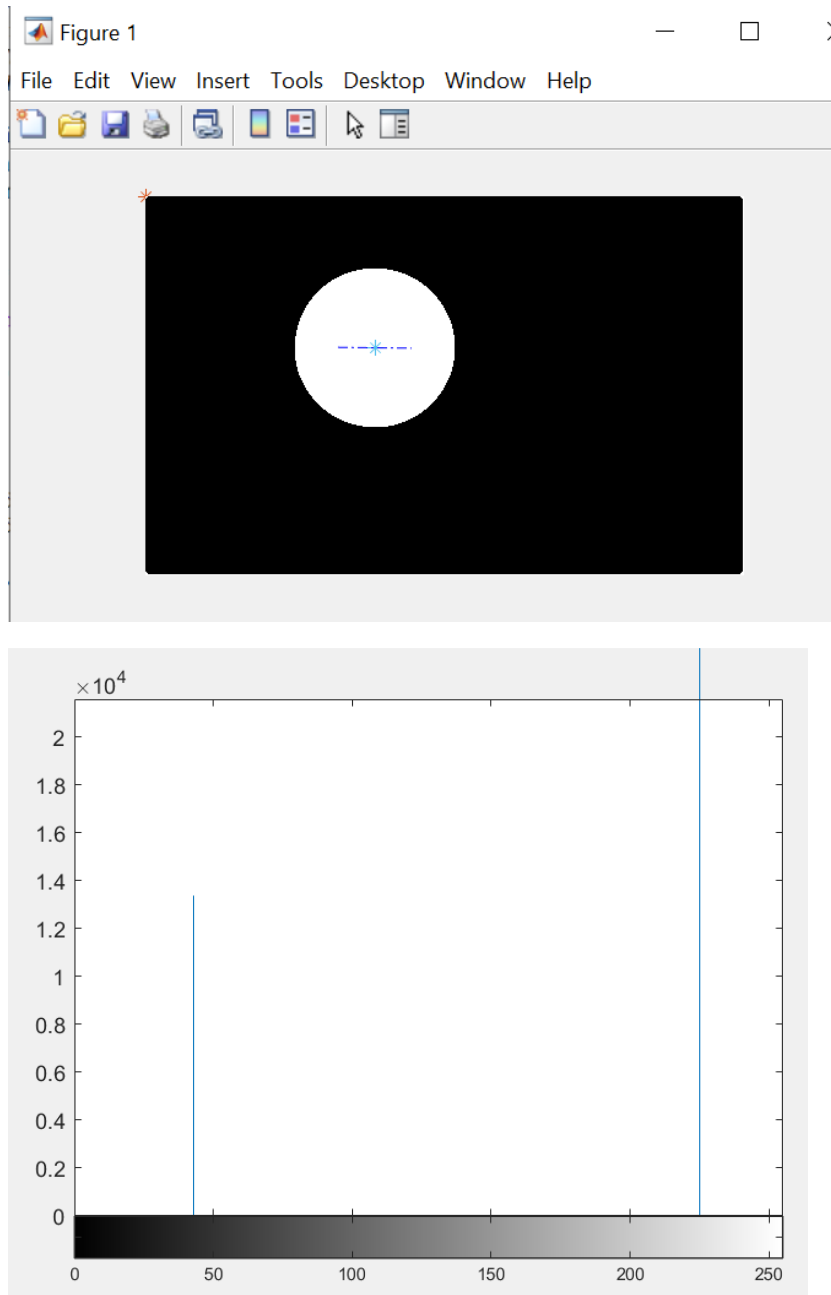
figure(1)
imshow(D)

for i=1:num
    centroids = cat(1, feats.Centroid);
    figure(1)
    imshow(D)
    hold on
    plot(centroids(:,1), centroids(:,2), 'b*')
    hold off
end

hold on
for k = 1:length(feats)
    x=feats(k).Centroid(1,1);
    y=feats(k).Centroid(1,2);
    figure(1)
    imshow(D)
    plot(x,y, '*')
    a1 = tan((feats(k).Orientation)*pi/180);
    b1 = y-a1*x;
    x1 = x-30;
    x2 = x+30;
    y1 = a1*x1+b1;
    y2 = a1*x2+b1;
    plot([x1 x2], [y1 y2], '-.b')
end
```

```
figure(2)
imhist(A)
hold off
```

Wynik:

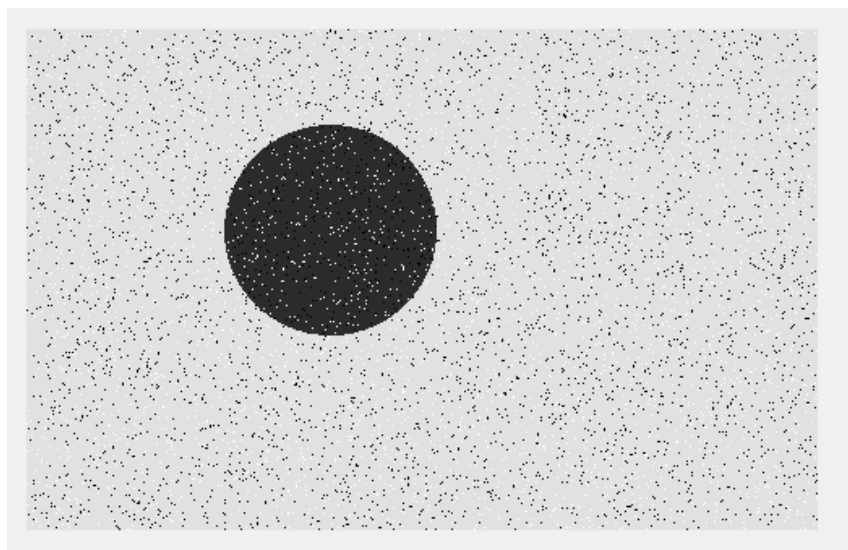


Wnioski:

Funkcja `regionprops` pokazuje różne cechy obrazu, jedną z nich – `Orientation` – używamy do ustalenia położenia obiektu na obrazie, ale dopiero po zbinaryzowaniu go. To znaczy kąta ustawienia względem osi układu odniesienia ustalonego na krawędziach obrazu.

## Zadanie 2 – Analiza obrazu: jeden obiekt – obraz zaszumiony szumem salt&pepper

Obraz:



Kod:

```
clc;
clear all;
close all;

info = imfinfo('ObrazZ1.png');

A = imread('ObrazZ1.png');
im_size = size(A);
A=rgb2gray(A);

hist = imhist(A);

E = imnoise(A, 'salt & pepper');

B = medfilt2(E, [8 8]);

hist2 = imhist(B);

T = 0.79;
C = im2bw(B,T);

D = not(C);

[num, L] = bwlabel(D);
feats = regionprops(D, 'all');

for i=1:num
    centroids = cat(1,
feats.Centroid);
    figure(1)
    imshow(D)

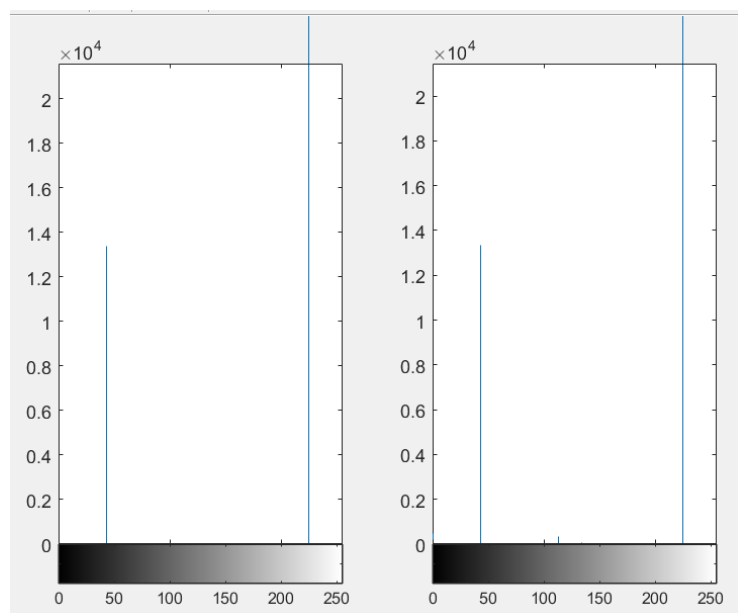
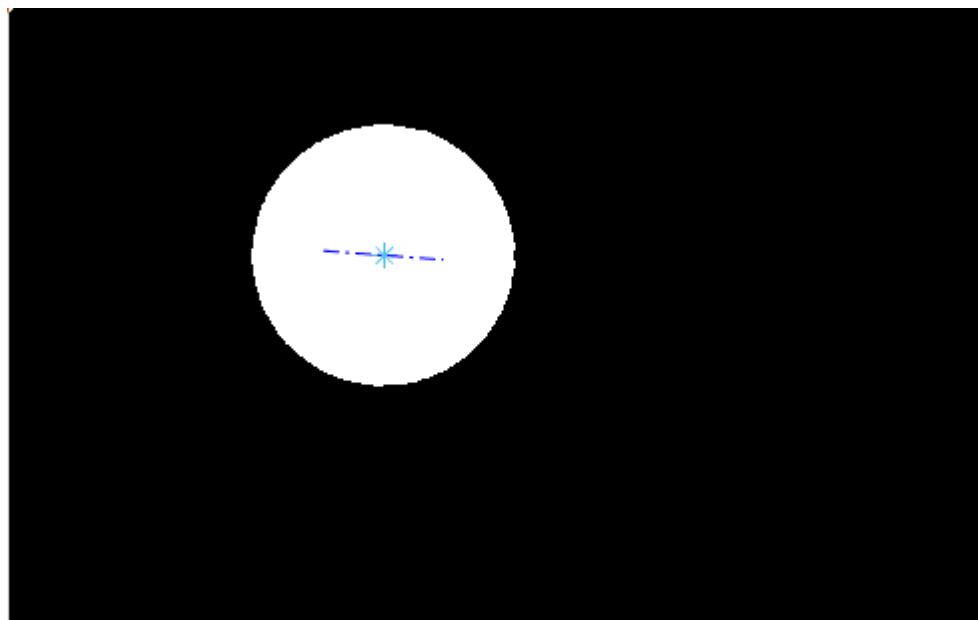
    hold on
    plot(centroids(:,1), centroids(:,2), 'b*')
    hold off
end

hold on
for k = 1:length(feats)
    x=feats(k).Centroid(1,1);
    y=feats(k).Centroid(1,2);
    figure(1)
    imshow(D)
    plot(x,y, '*')
    a1 = tan((feats(k).Orientation)*pi/180);
    b1 = y-a1*x;
    x1 = x-30;
    x2 = x+30;
    y1 = a1*x1+b1;
    y2 = a1*x2+b1;
    plot([x1 x2], [y1 y2], '-.b')
end
hold off

figure(2)
subplot(121)
imhist(A)
subplot(122)
imhist(B)

figure(3)
imshow(E)
```

Wyniki:



Wnioski:

By ustalić położenie obiektu na obrazie oprócz wcześniejszej binaryzacji należy usunąć szumy filtrami, ponieważ będą zakłócały w znalezieniu obiektu na obrazie.

Zadanie 3 – Analiza obrazu: wybierz jeden obiekt spośród 9 obiektów

Obraz:



Kod:

```
clc;
clear all;
close all;

info = imfinfo('ObrazZ3pop.png');

A = imread('ObrazZ3pop.png');
im_size = size(A);
A=rgb2gray(A);

hist = imhist(A);

E = imnoise(A, 'salt & pepper');

B = medfilt2(E, [8 8]);

feats = regionprops(D, 'all');
figure(1)
imhist(B);

fil=(A>=40) & (A<=45);

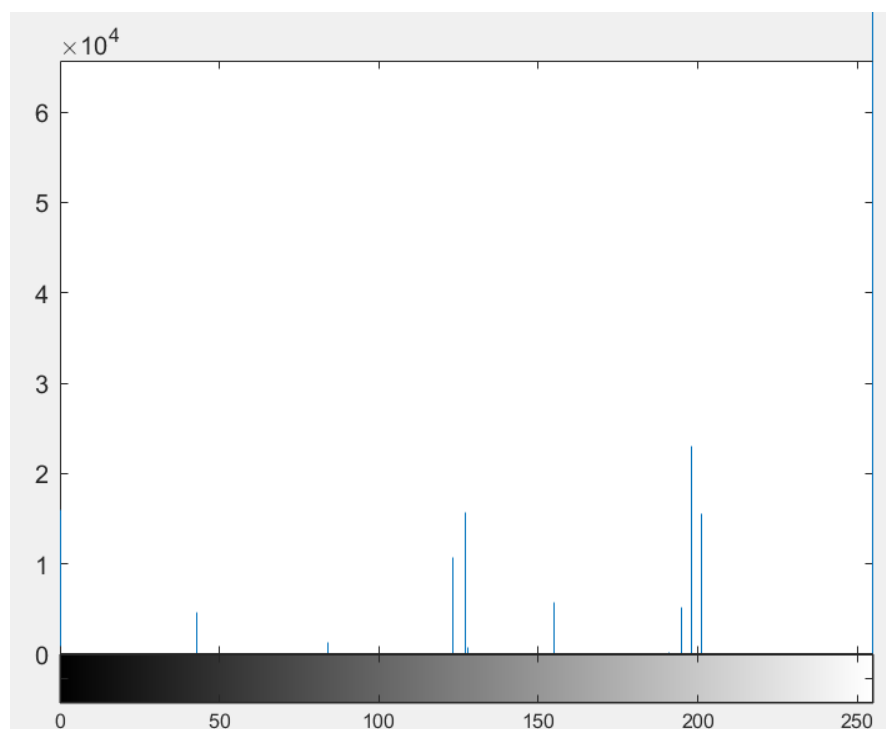
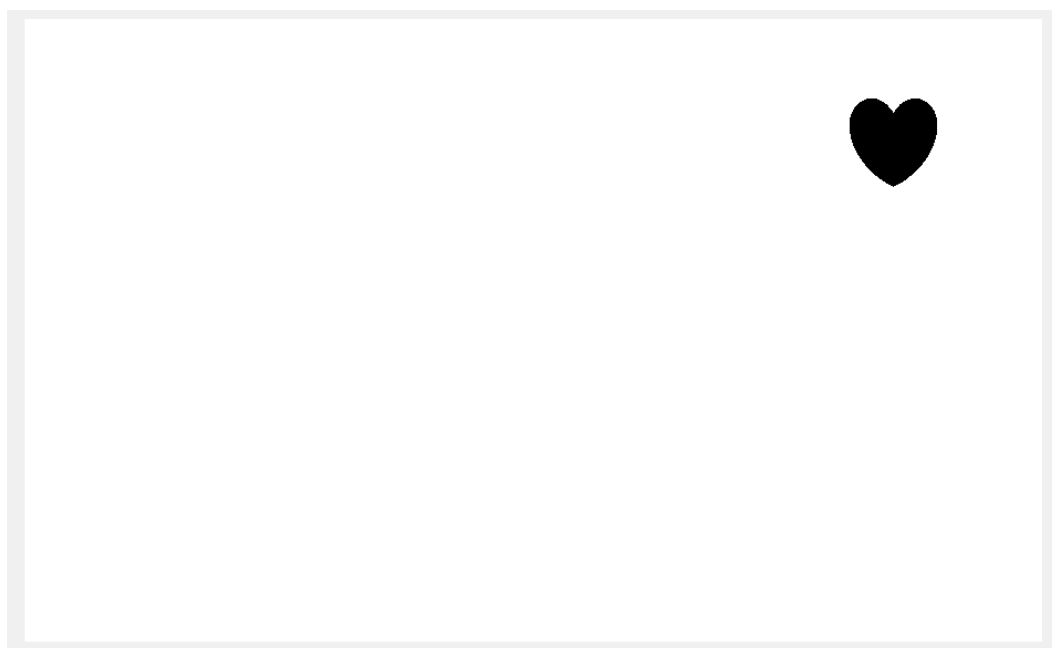
T = 0.79;
C = im2bw(fil,T);

D = not(C);

figure(2)
imshow(D)

[num, L] = bwlabel(D);
```

Wyniki:

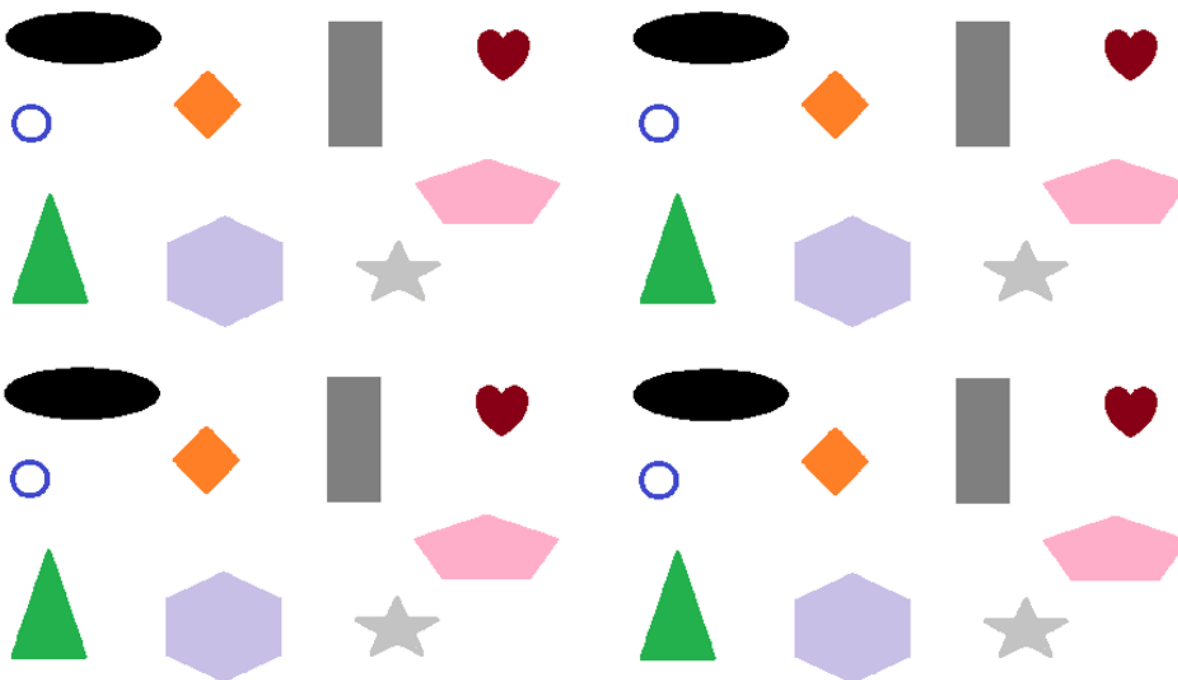


Wnioski:

Dzięki funkcji regionprops odczytujemy właściwości kolejnych obiektów uporządkowanych za pomocą funkcji bwlabel. Odczytałam i ograniczyłam wartości skali szarości tak by pokazać tylko wybrany przeze mnie obiekt.

#### Zadanie 4 – Analiza obrazów: współczynniki kształtu dla 9 analizowanych obiektów

Obraz:



Kod:

```
close all;
clear all;
clc;

imfinfo('ObrazZ4.png')
A=imread('ObrazZ4.png');
B=rgb2gray(A);

figure(1)
imhist(B)
T = 0.79;
C = im2bw(B,T);
im_bw = not(C);
figure(2)
imshow(im_bw)
hold on

[L,num]=bwlabel(im_bw,4);
feats=regionprops(L,'all');

for i=1:num
    centroids = cat(1, feats.Centroid);

form_factor(i)=cat(1,(4*pi*feats(i).Area)/(feats(i).Perimeter*feats(i).Perimeter));
```

```

roundness(i)=cat(1,(4*feats(i).Area)/(pi*feats(i).MajorAxisLength*feats(i).MajorAxisL
ength));
    aspect_ratio(i)=cat(1,feats(i).MajorAxisLength/feats(i).MinorAxisLength);
    solidity(i)=cat(1,feats(i).Area/feats(i).ConvexArea);
    compactness(i)=cat(1,(sqrt(4*(feats(i).Area/pi))/feats(i).MajorAxisLength));
    plot(centroids(:,1), centroids(:,2), 'b*')

end
for k=1:length(feats)
    x=feats(k).Centroid(1,1);
    y=feats(k).Centroid(1,2);
    plot(x,y, '*')
end
hold off

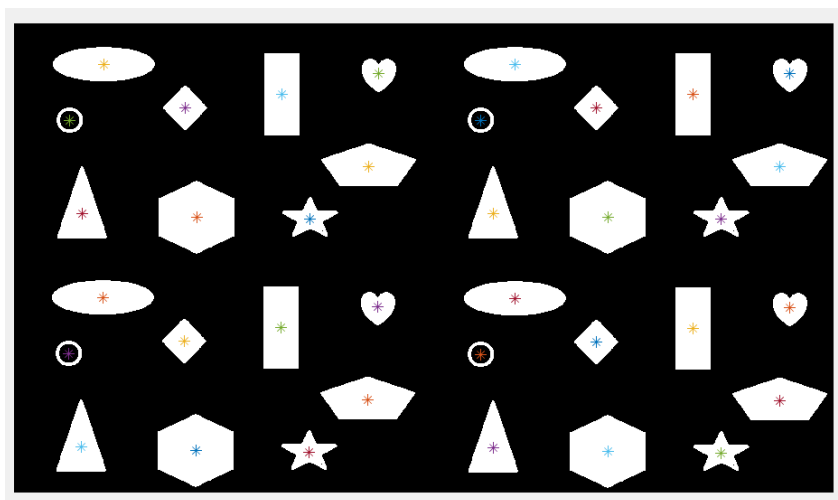
E = bwconncomp(im_bw);
L = labelmatrix(E);

figure(3)
F=find([roundness] <= 0.3);
G = ismember(L, F);
imshow(G)
hold on

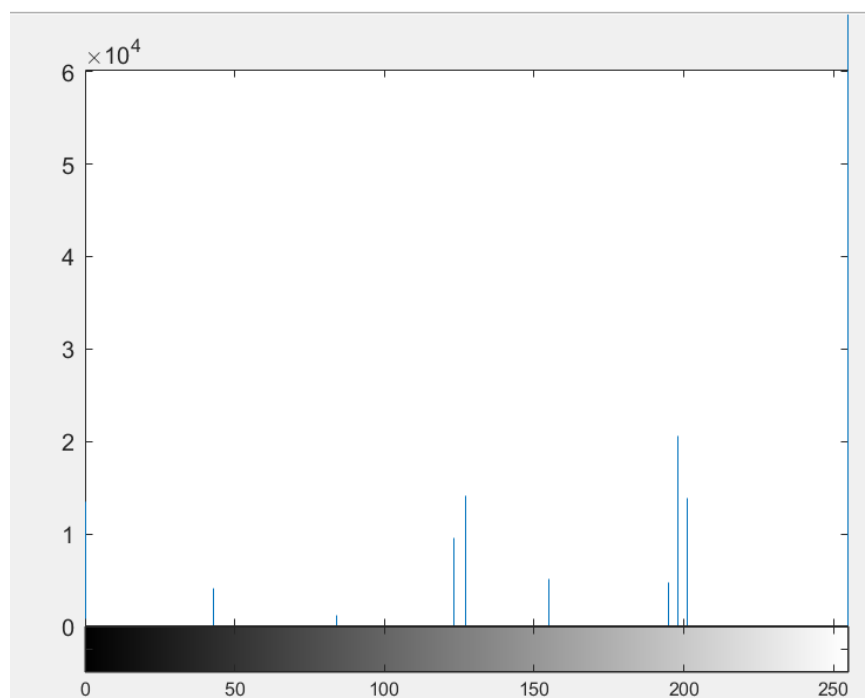
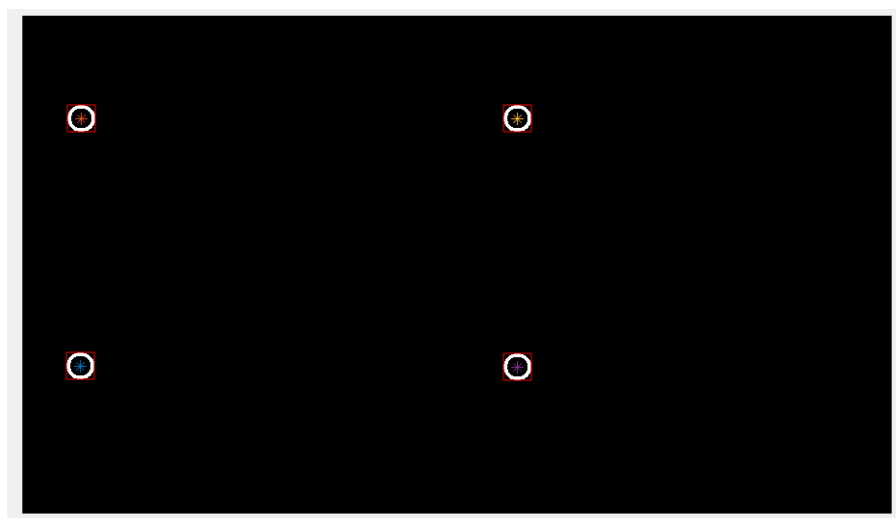
for k=1:length(feats)
    if roundness(k) <= 0.3
        rectangle('Position', feats(k).BoundingBox, 'EdgeColor', 'red');
        x=feats(k).Centroid(1,1);
        y=feats(k).Centroid(1,2);
        plot(x,y, '*')
    end
end
hold off

```

Wyniki:







Zadanie 5 – Analiza obrazów: statystyka dla wybranych cech obiektów

Przeznaczona cech orientacji

Obraz:



Kod:

```
clc;
```

```
imfinfo('rice.png.jpg')
A=imread('rice.png.jpg');
B=rgb2gray(A);
T = 0.5;
D = medfilt2(B, [8 8]);
C = im2bw(D,T);
figure(1)
imhist(B)
figure(2)
subplot(121)
imshow(A)
subplot(122)
imshow(C)

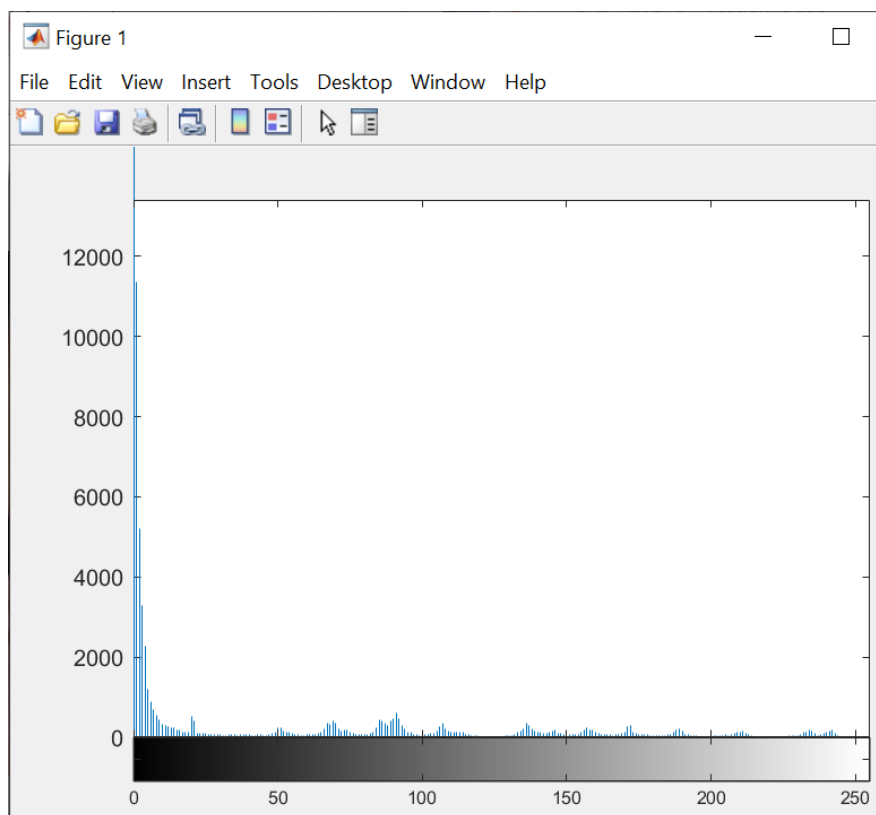
se = strel('disk',15);
background = imopen(A,se);
% I2 = A - background;
% I3 = imadjust(I2);
% im_bw = imbinarize(I3);
% im_bw = bwareaopen(C,50);
figure(4)
imshow(C)
[L,n]=bwlabel(C,4);
feats=regionprops(L,'all');
Array = [feats.Orientation];
%Array = find (Array1 > 0.94);
```

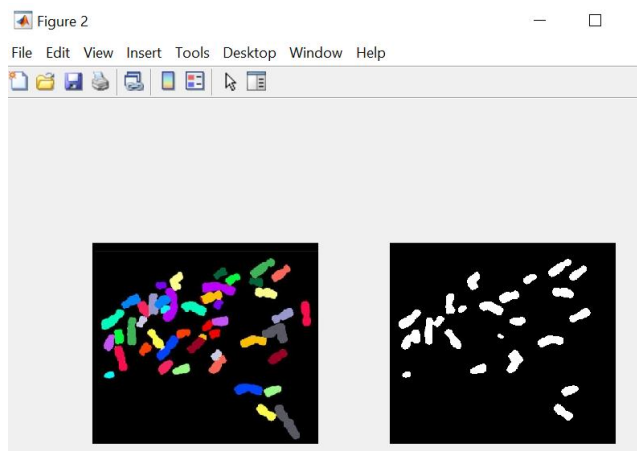
```

Min = min(Array);
Max = max(Array);
Mean = mean(Array);
Std = std(Array);
figure('Name', 'histogram for orientation'), hist(Array, 20);
idx = find(Array < (Mean-Std));
L2 = ismember(L, idx);
L2 = 1.*L2;
figure('Name', 'Objects below mean-std: orientation'), imshow(L2);
idx = find(Array > (Mean+Std));
L2 = ismember(L, idx);
L2 = 1.*L2;
figure('Name', 'Objects above mean+std: orientation'), imshow(L2);
idx = find((Array < (Mean+Std)) & (Array > (Mean-Std)));
L2 = ismember(L, idx);
L2 = 1.*L2;
figure('Name', 'Objects between mean-std and mean+std: orientation'), imshow(L2);

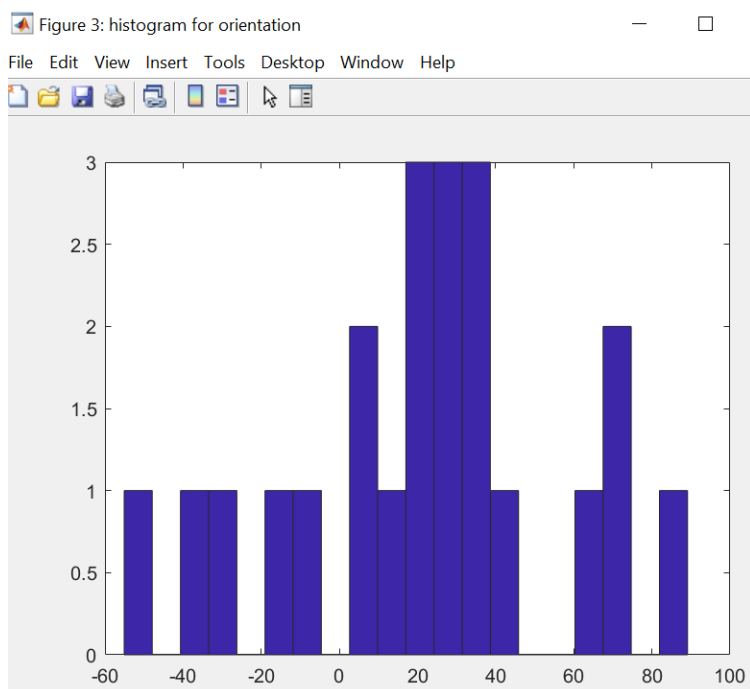
```

Wyniki:





Postanowiłam nie wyznaczać wszystkich obiektów z obrazka (by ukazać więcej należytny zmniejszyć treshold), ponieważ w ten sposób lepiej widać wyniki zadania dla cechy orientation.



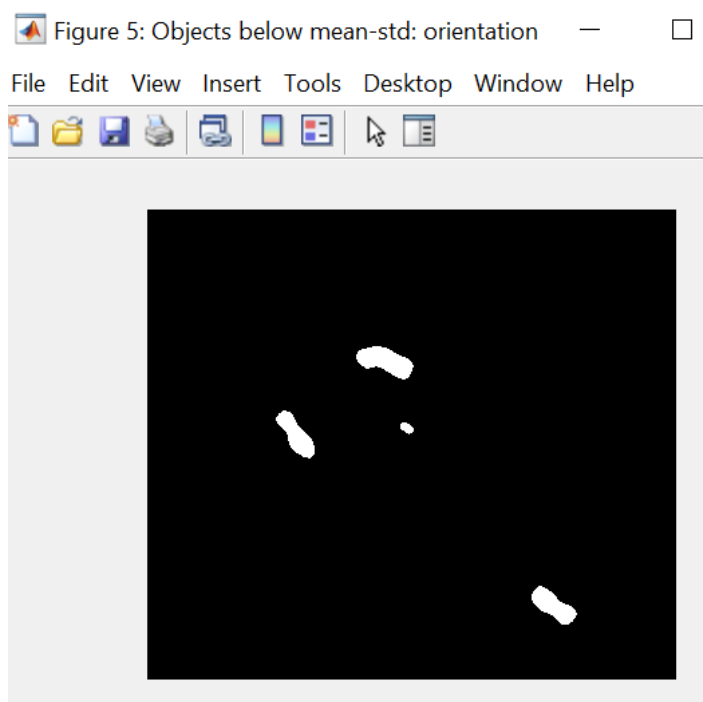
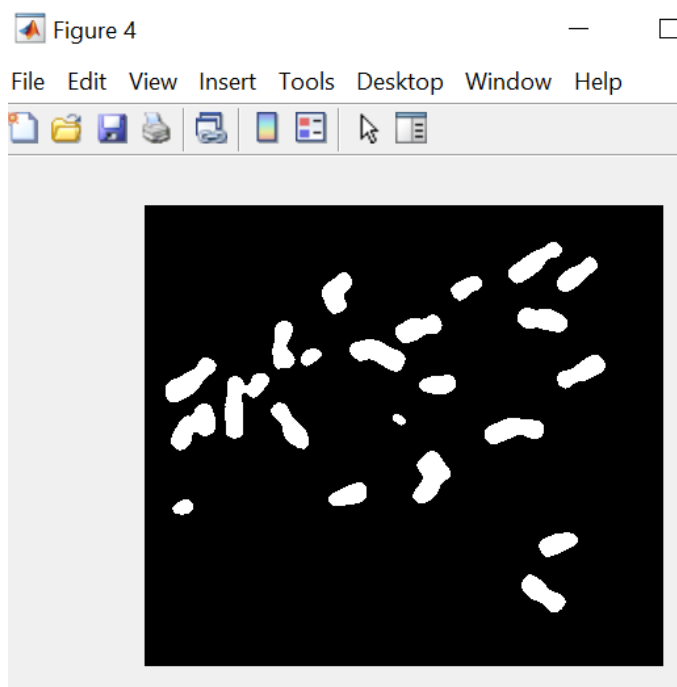


Figure 6: Objects above mean+std: orientati...

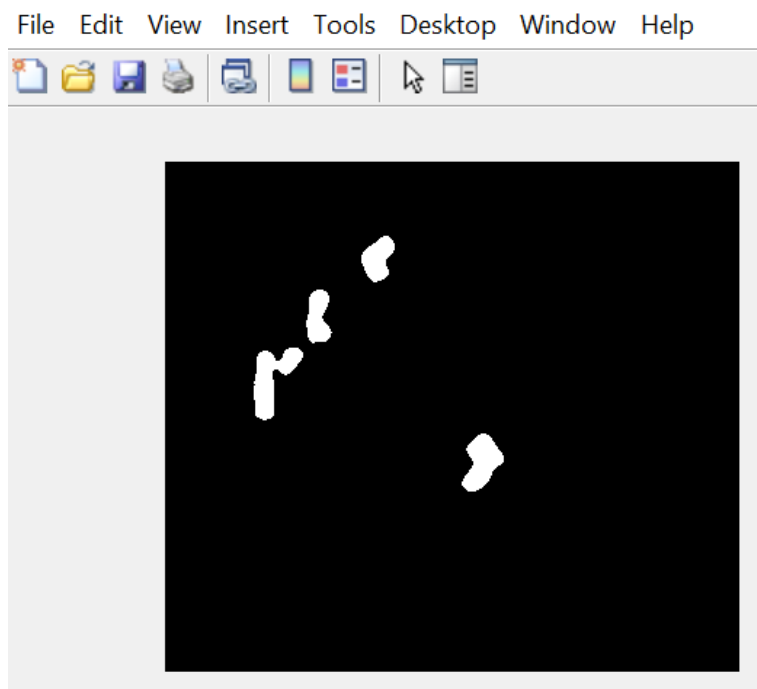
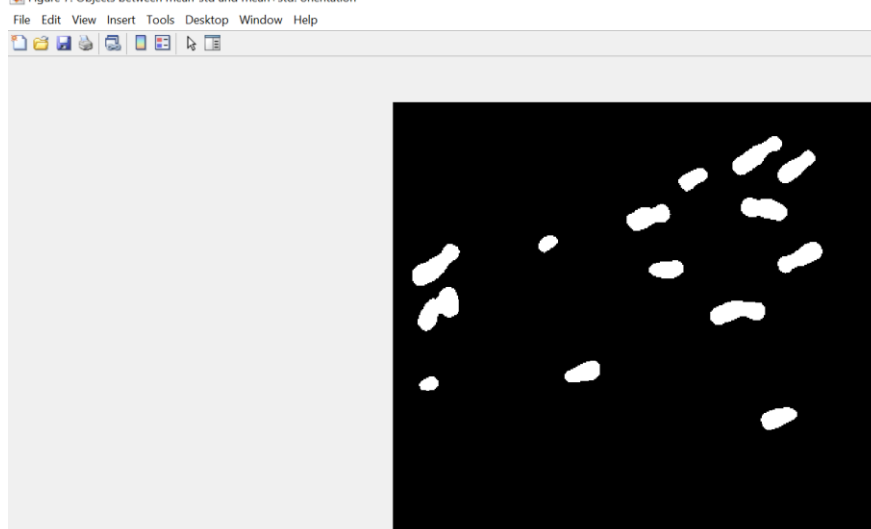


Figure 7: Objects between mean-std and mean+std: orientation



## Zadanie 6 – Analiza obrazów: Momenty inwariantne

Kod:

```
clear all;
close all;
clc;

A=imread('ObrazZ1.png');
B=rgb2gray(A);
oryginal = B;
translacja = imtranslate(oryginal,[15, 25],'OutputView','full');
odbicie = flip(oryginal,2);
rozmiar = imresize(oryginal,2);
rotacja45 = imrotate(oryginal, 45);
rotacja90 = imrotate(oryginal, 90);
figure(1);
subplot(2,3,1);
imshow(oryginal);
title('Obraz oryginalny');
subplot(2,3,2);
imshow(translacja);
title('Obraz przesuniety');
subplot(2,3,3);
imshow(odbicie);
title('Odbicie lustrzane');
subplot(2,3,4);
imshow(rozmiar);
title('Obraz zmniejszony');
subplot(2,3,5);
imshow(rotacja45);
title('Obraz po rotacji o 45deg');
subplot(2,3,6);
imshow(rotacja90);
title('Obraz po rotacji o 90deg');
phioryginal = abs(log(invmom(oryginal)));
phiobrot = abs(log(invmom(translacja)));
phiodbicie = abs(log(invmom(odbicie)));
phirozmiar = abs(log(invmom(rozmiar)));
phirotacja45 = abs(log(invmom(rotacja45)));
phirotacja90 = abs(log(invmom(rotacja90)));
E = medfilt2(A, [8 8]);

T = 0.5;
C = im2bw(E,T);

D = not(C);
figure(2)
imshow(D)
[L,n]=bwlabel(D,4);
feats=regionprops(L,'all');
% Współczynnik Malinowskiej
for i=1:n
    Ob=feats(i).Perimeter;
    S=feats(i).Area;
```

```

WM(1,i)=(Ob/(2*((pi*S)^(0.5))))-1;
end
WM
% Współczynnik cyrkularności 1 - WC1
for i=1:n
    Pole = feats(i).Area;
    WC1(1,i) = 2*(sqrt(Pole/pi));
end
WC1
% Współczynnik cyrkularności 2 - WC2
for i=1:n
    Ob = feats(i).Perimeter;
    WC2(1,i) = Ob/pi;
end
WC2
% Współczynnik Fereta
for i=1:n
    Dh = feats(i).BoundingBox(3);
    Dv = feats(i).BoundingBox(4);
    WF(1,i) = Dh/Dv;
end
WF

```

Wyniki:

