



**Politechnika  
Śląska**

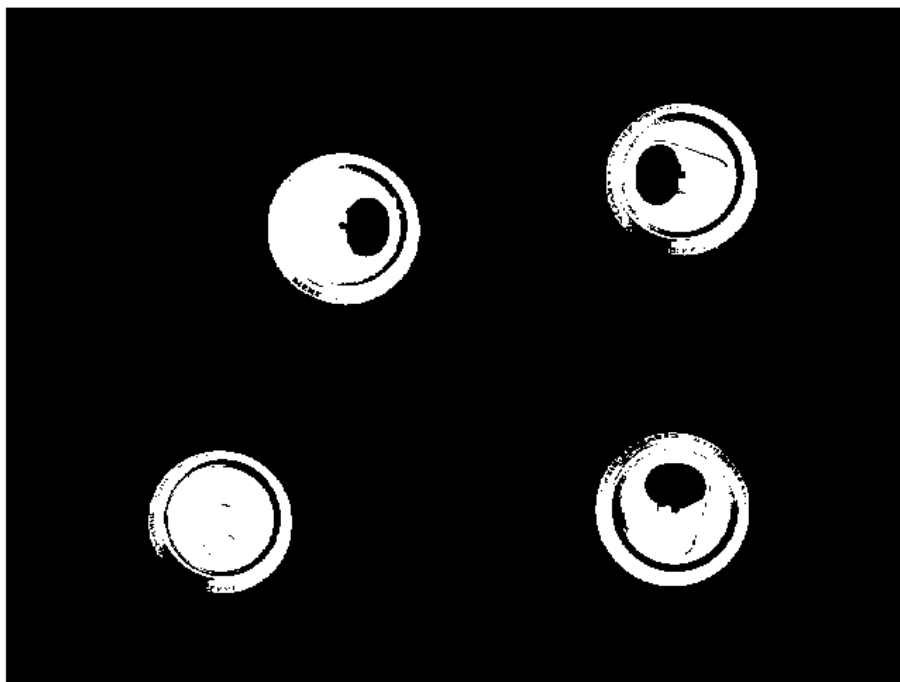
**Politechnika Śląska w Gliwicach**  
**Wydział Automatyki, Elektroniki i Informatyki**  
Przetwarzanie obrazów cyfrowych

**STUDIUM PRZYPADKU – DETEKCJA I  
WYZNACZANIE CECH OBIEKTÓW**

# 1 Opis działania algorytmu

## 1.1 Krok 1

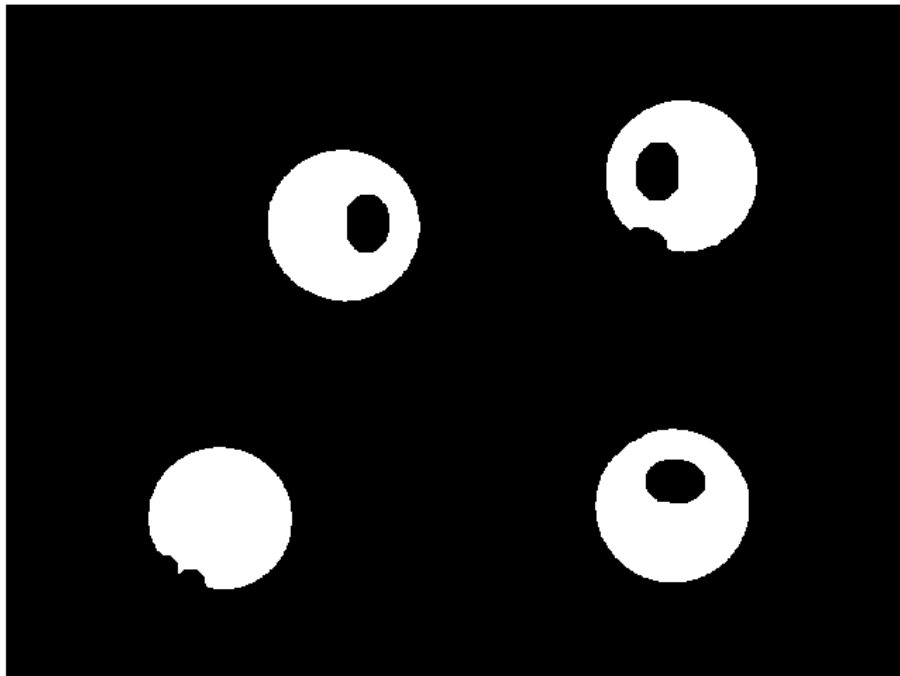
Pierwszym krokiem jest sprawdzenie czy obraz jest kolorowy jeśli tak następuje zamiana na obraz w skali szarości. Następnym krokiem jest binaryzacja obrazu.



Rysunek 1: Obraz po binaryzacji

## 1.2 Krok 2

Kolejnym krokiem jest zamknięcie obrazu w celu pozbycia się dziur. Kształt obiektów może się lekko zdegradować lecz nie wpłynie to na poprawność działania programu.



Rysunek 2: Obraz po operacji zamykania

### 1.3 Krok 3

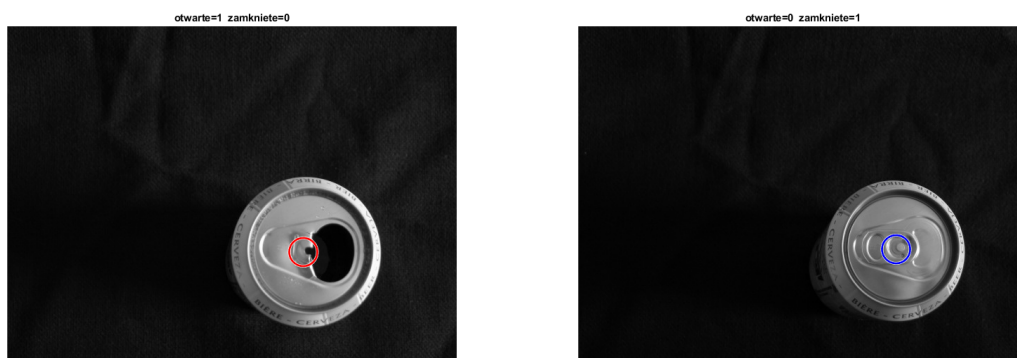
Po wykonaniu zamykania obraz jest etykietowany funkcją `bwlabel` w celu policzenia ilości obiektów oraz funkcją `regionprops` obliczane są parametry obrazu takie jak liczba Eulera oraz inne statystyki obrazu. Na podstawie liczby Eulera określone jest czy puszka jest zamknięta czy otwarta, jeśli puszka jest otwarta liczba dziur w obiekcie wynosi 1, natomiast jeśli zamknięta 0. Następnie zaznaczane są puszki otwarte oraz zamknięte na oryginalnym obrazie za pomocą okręgów, niebieskich jeśli puszka jest zamknięta oraz czerwonych gdy otwarta.



Rysunek 3: Wynik operacji

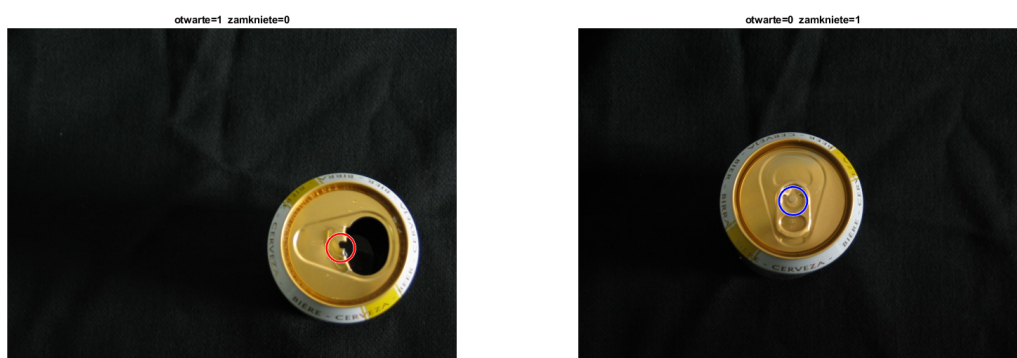
## 2 Prezentacja wyników

### 2.1 Obrazy z jednym obiektem w skali szarości



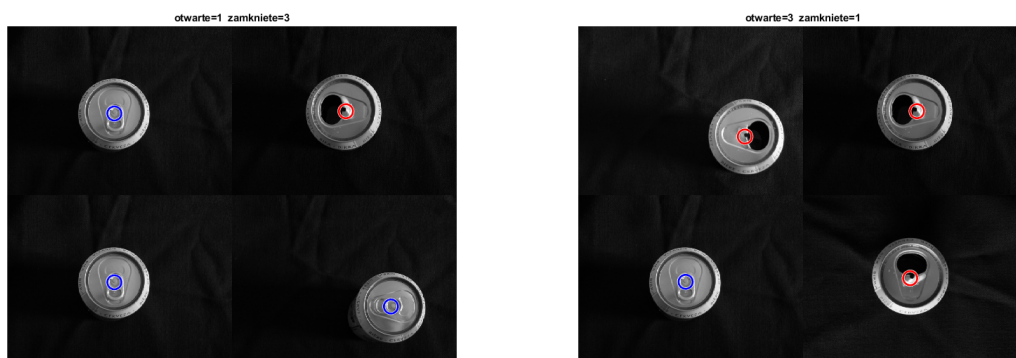
Rysunek 4: Obrazy w skali szarości

### 2.2 Obrazy z jednym obiektem w RGB



Rysunek 5: Obrazy w RGB

## 2.3 Obrazy z wieloma obiektami



Rysunek 6: Obrazy z wieloma obiektami

## 3 Wnioski

Algorytm poradził sobie poprawnie z wszystkimi problemami konieczne było jednak zastosowanie operacji morfologicznej zamykania obrazu. Niektóre obiekty okazały się zniekształcone co nie było problemem przy zastosowaniu zliczania dziur.

## 4 Kod z programu matlab

```
1 clear all
2 imstruct.ori=imread("obrazy2\test2.bmp");
3 size1=size(imstruct.ori,3);
4 if size1==3
5     imstruct.ori=rgb2gray(imstruct.ori);
6 else
7     imstruct.ori=imstruct.ori;
8 end
9
10 % imshow(imstruct.img)
11
12
13
14
15 imstruct.img=imbinarize(imstruct.ori,'global');
16 imshow(imstruct.img)
17
18 se = strel('disk',20);
19 imstruct.closeBW = imclose(imstruct.img,se);
20 figure , imshow(imstruct.closeBW)
21
22 [imstruct.label , imstruct.nb_of_ob]=bwlabel(imstruct.closeBW);
23 imstruct.stats=regionprops("table",imstruct.closeBW,"Perimeter","Area",
    "Centroid","EulerNumber");
24 imstruct.eulernb=uint8((imstruct.stats.EulerNumber-1)*(-1))
25
26
27 imshow(imstruct.ori)
28
29 zamkniete=0;
30 otwarte=0;
31
32 for i=1:imstruct.nb_of_ob
33
34     if imstruct.eulernb(i)==0
35
36         viscircles(imstruct.stats.Centroid(i,:),20,'color','b');
37         zamkniete=zamkniete+1;
38     else
39         viscircles(imstruct.stats.Centroid(i,:),20,'color','r');
40         otwarte=otwarte+1;
41     end
42     title(['otwarte=' , num2str(otwarte) , ' zamkniete=' , num2str(
        zamkniete)])
43 end
```