University of Belgrade - Faculty of Electrical Engineering

Department of Signals and Systems

13E054DOS – Digital image processing assignment

Student: Teodora Spasojević

Index count: 0427/2019

February 2022.

Task text

II Domaći zadatak iz Digitalne Obrade Slike - Zadatak br. 14

-

Slika 216 prikazuje zrna kafe. Zadatak je da se izdvoje zrna nadprosečne veličine. Originalnu sliku treba filtrirati a zatim segmentisati proizvoljnim izborom praga sjajnosti. Morfološkim obradama treba u binarnoj slici ukloniti ivične objekte, a zatim izvršiti razdvajanje spojenih. Za preostalu populaciju zrna izračunati srednju vrednost površine, izdvojiti nadprosečne, odrediti njihove centroide i superponirati ih na originalnu sliku.

Theoretical introduction

Morphological processing provides us with a set of tools that, above all, we use when processing binary images (images where we have only white or black, not shades of gray). Morphological processing improves the quality of images, and the reason we use it primarily is to better show objects of interest in the image.

Some of the tools we use in morphological image processing are erosion, dilation, opening, closing, and hit or miss transformations. By using these tools in the image, we can perform many significant corrections, such as extracting boundaries, filling regions, bolding and thinning shapes, creating skeleton sculpture and so on.

Erosion is a set of shifts, z, such that the structuring element, B , transliterated for z, and set A, overlap in all elements.

$$A \ominus B = \{z \mid (B)_z \subseteq A\}$$

Due to the requirement that the structuring element must be folded over all elements with pixels of shape, erosion leads to a shrinking shape. It is used when we want to eliminate irregularities at the edges of a shape and the connection between shapes.

Dilation is a set of shifts, z, for which the reflection of a structuring element, B, and the set A, overlap at least in one element.

$$A \oplus B = \left\{ z \mid \left(\hat{B} \right)_z \cap A \neq \emptyset \right\}$$

$$A \oplus B = \left\{ z \mid \left[\left(\hat{B} \right)_z \cap A \right] \subseteq A \right\}$$

Due to the requirement that the structuring element must be folded at least once by the element in with the pixels of the shape, the dilatation leads to the expansion of the shape. It is used when we want to fill cavities and various interruptions in contours.

Opening is the combined use of erosion, then dilation.

$$A \circ B = \left(A \ominus B \right) \oplus B$$

Opening leads to erosion effects, such as ironing contours and eliminating thin protrusions.

Closure is the combined use of dilation, then erosion.

$$A \bullet B = \left(A \oplus B \right) \ominus B$$

Closing leads to dilation effects, such as ironing parts of the contours, filling cavities, filling breaks in the contour, and joining the break.

A hit or miss transformation is a transformation based on the use of two structuring elements. The first describes the structure of the shape we want to guess, and the second describes the environment of the object that we do not want to encompass. By using these two structuring elements, we are sure that we will be able to locate the position of the desired shape, and that we will avoid all forms of the same structure, but a little larger dimensions.

For structuring element B1 we take the appearance of the desired shape: B1 = X.

For structuring element B2 we take the difference between window W and desired shape: B2 = W -X.

The hit or miss transformation consists of the following steps:

1. The first step is to compute the erosion of the set of all forms A with the structuring element B1.
2. The second step is to compute the complement erosion of a set of all forms A^c with the structuring element B2.
3. The third step is to find the intersection of the results of the two obtained erosions.

This transformation at the output yields a pixel that indicates the center of the requested shape.

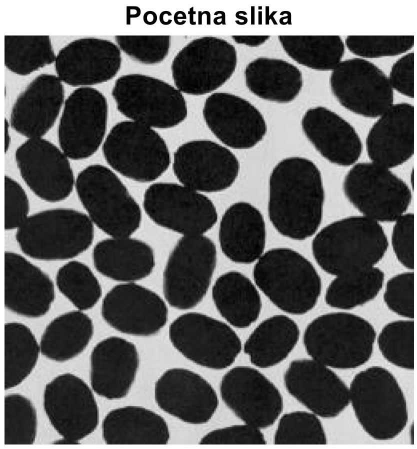
Solution

Code:

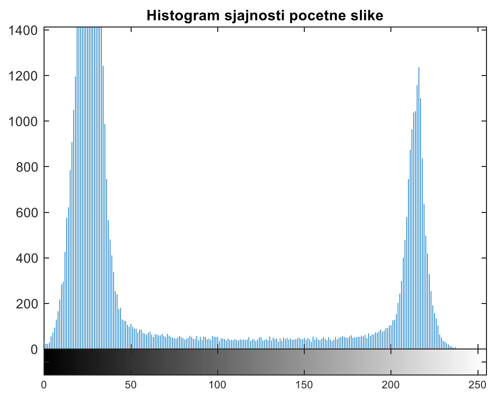
```
%% Domaci zadatak br. 2

clear;
close all;
clc;

%% Ucitavanje i prikaz slike

img = imread('slika 216.jpg');

figure, imshow(img), title('Pocetna slika')

%% Segmentacija slike koriscenjem proizvoljnog praga sjajnosti

figure, imhist(img), title('Histogram sjajnosti pocetne slike')

T = 125;
img_bin = img >= T;
figure, imshow(img_bin), title('Binarizaovana slika')
imwrite(img_bin, 'Binarizovana slika.jpg');

%% Razdvajanje spojenih objekata

img_bin_neg = imcomplement(img_bin);
figure, imshow(img_bin_neg), title('Negativ binarizovane slike')

se = strel('disk', 10);
img_separated_beans_neg = imerode(img_bin_neg, se);
figure, imshow(img_separated_beans_neg), title('Slika posle erozije')

se = strel('disk', 8);
img_separated_beans_neg = imdilate(img_separated_beans_neg, se);
figure, imshow(img_separated_beans_neg), title('Slika posle erozije i dilatacije')
```

```
img_separated_beans = imcomplement(img_separated_beans_neg);
figure, imshow(img_separated_beans), title('Slika sa razdvojenim zrnima')
imwrite(img_separated_beans, 'Slika sa razdvojenim zrnima.jpg');


%% Uklanjanje ivicnih objekata iz slike

img_no_border_beans_neg = imclearborder(img_separated_beans_neg);
img_no_border_beans = imcomplement(img_no_border_beans_neg);
figure, imshow(img_no_border_beans), title('Slika bez ivicnih zrna')
imwrite(img_no_border_beans, 'Slika bez ivicnih zrna.jpg');


img_final_neg = img_no_border_beans_neg;
img_final = img_no_border_beans;
%% Racunanje srednje vrednosti povrsine zrna

[L, num] = bwlabel(img_final_neg);

total_surface = 0;

for k=1:num

    current_surface = 0;
    object = find(L == k);
    curr_surface = length(object);
    text = ['Povrsina ', num2str(k), '. oblika je ', num2str(curr_surface), ' piksela.'];
    disp(text);
    total_surface = total_surface + curr_surface;

end

total_surface = total_surface/num;
text = ['Prosecna povrsina oblika je ', num2str(total_surface), ' piksela.'];
disp(text);
%% Izdvajanje natprosecno velikih zrna

se = strel('disk', 13);

img_find_biggest_beans = imerode(img_final_neg, se);
figure, imshow(img_find_biggest_beans), title('Pozicije najvecih zrna')

img_biggest_beans_neg = imreconstruct(img_find_biggest_beans, img_final_neg);
img_biggest_beans = imcomplement(img_biggest_beans_neg);
figure, imshow(img_biggest_beans), title('Slika sa najvecim zrnima')
imwrite(img_biggest_beans, 'Slika sa najvecim zrnima.jpg');
%% Nalazenje i crtanje centroida zrna

figure, imshow(img_final);
hold on

[L, num] = bwlabel(img_final_neg);

for k=1:num

    [r, c] = find(L == k);
    rbar = mean(r);
    cbar = mean(c);
    plot(cbar, rbar, 'Marker', 'o', 'MarkerEdgeColor', 'w', ...
        'MarkerFaceColor', 'w', 'MarkerSize', 10)
    plot(cbar, rbar, 'Marker', '*', 'MarkerEdgeColor', 'k')

end

title('Slika zrna sa pozicijama njihovih centroida')
hold off
```

Results

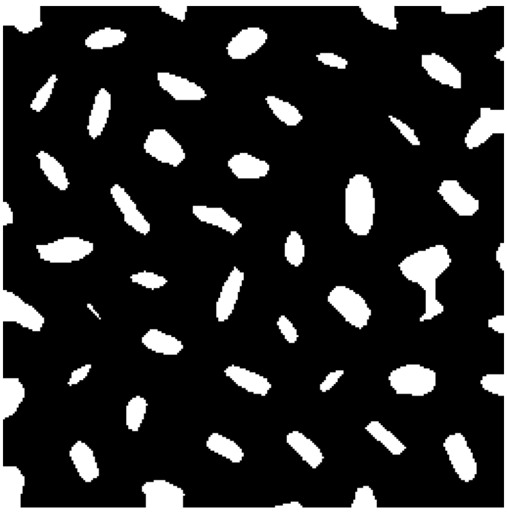


In the picture there are thickened coffee beans. To begin, it is necessary to transfer the image from the grey scale to the binary image. We do this by segmenting the image using the threshold of brilliance. When the histogram of an image is plotted, we see that it is ok for the brightness threshold to use the value of 125. In segmentation, for the values of all pixels whose shades of gray are less than 125, we will enter the value of 0 (black color), and otherwise the value of 1 (white).

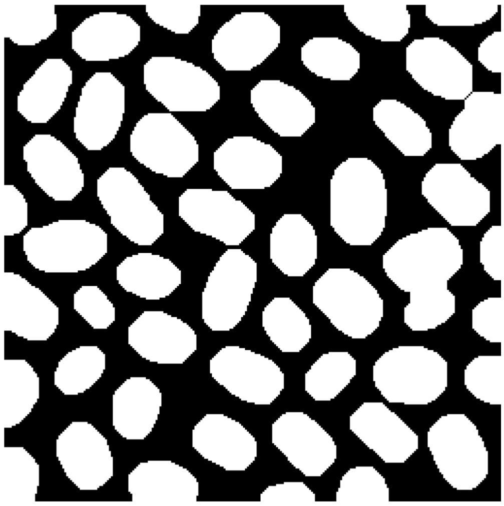


Then it is necessary to separate the coffee beans from each other. We do this so that each grain from the picture can be recognized as a separate shape. To separate the coffee beans in the picture, we first apply erosion. Then we get a picture where we see traces of where the grains are. Then we use the dilation, in order to maximize the dimension of the grains, without them connecting again. We did not use the built-in opening function, because we adjusted the dimensions of the structuring element separately during erosion and dilation. For the shape of the structuring element we chose a disc, because it best suits our shapes.

Slika posle erozije

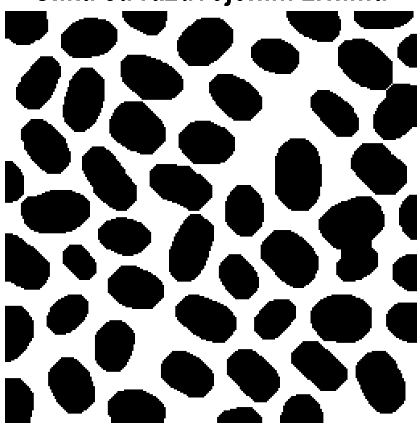


Slika posle erozije i dilatacije

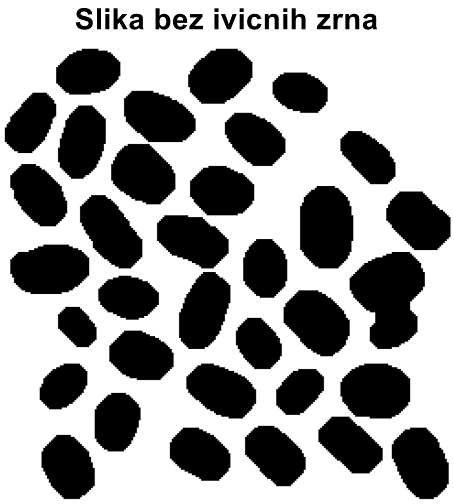


These operations give better results if used on the negative of our images. That's why we these results in the process. As a result of the separation of the grain, we get the following picture:

Slika sa razdvojenim zrnima



Next, we want to remove the shapes that are on the edge of the image, because they are not representative in shape and size. This is done by using the imclearborder function.



When we are fully finished with processing the image, we can move on to extracting certain shapes from the image and extracting different features from the form.

In order to calculate the grain surfaces from the image, we use the bwlabel function, which recognizes the shapes in the image, according to some of the rules of adjacentness, and returns the number of those shapes and the matrix with the inscribed indexes of shapes on the pixels where those shapes are located. If we go through the for and count the number of pixels that belong to each shape, the following results are obtained:

The surface of the 1st shape is 634 pixels.

The surface of the 2nd shape is 736 pixels.

The surface of the 3rd shape is 950 pixels.

The surface of the 4th shape is 485 pixels.

The surface of the 5th shape is 764 pixels.

The surface of the 6th shape is 700 pixels.

The surface of the 7th shape is 797 pixels.

The surface of the 8th shape is 337 pixels.

The surface of the 9th shape is 875 pixels.

The surface of the 10th shape is 645 pixels.

The surface of the 11th shape is 601 pixels.

The surface of the 12th shape is 732 pixels.

The surface of the 13th shape is 867 pixels.

The surface of the 14th shape is 840 pixels.

The surface of the 15th shape is 810 pixels.

The surface of the 16th shape is 706 pixels.

The surface of the 17th shape is 866 pixels.

The surface of the 18th shape is 806 pixels.

The surface of the 19th shape is 805 pixels.

The surface of the 20th shape is 760 pixels.

The surface of the 21st shape is 693 pixels.

The surface of the 22nd shape is 523 pixels.

The surface of the 23rd shape is 627 pixels.

The surface of the 24th shape is 754 pixels.

The surface of the 25th shape is 529 pixels.

The surface of the 26th shape is 483 pixels.

The surface of the 27th shape is 930 pixels.

The surface of the 28th shape is 1111 pixels.

The surface of the 29th shape is 711 pixels.

The surface of the 30th shape is 581 pixels.

The surface of the 31st shape is 945 pixels.

The surface of the 32nd shape is 1475 pixels.

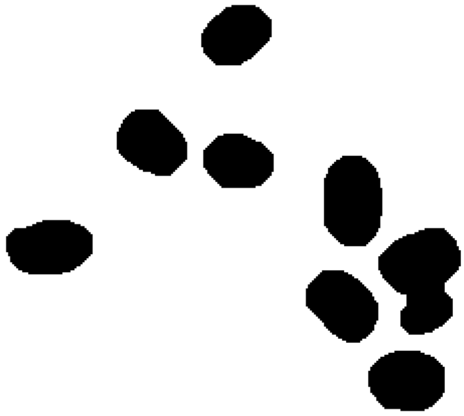
The surface of the 33rd shape is 822 pixels.

The surface of the 34th shape is 900 pixels.

The average surface of the shape is 758.8235 pixels.

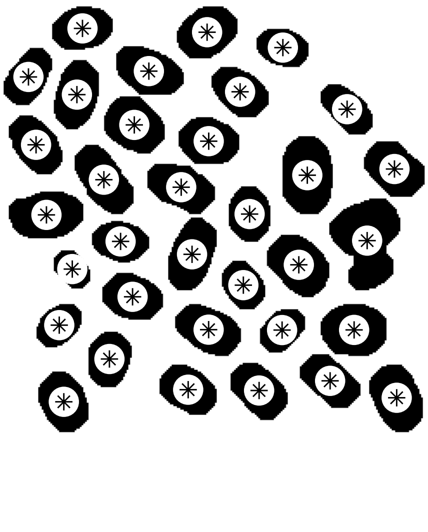
Then from the picture we can extract above-average large grains. We achieve this by using erosion with structuring elements that will only overlap with above-average grains. If we reconstruct the image with the positions of large grains drawn, using the function of imreconstruct, based on the image with all the grains, and we get an image with above-average large grains.

Pozicije najvecih zrna



Slika sa najvecim zrnima

The last requirement is to draw the centroids of all of the grains from the picture. We do this by going through all of the shapes recognized by the bwlabel function again and calculate the mean values for the position of pixels on the x and y axes of each shape. With this procedure, we get the following results:



Slika zrna sa pozicijama njihovih centroida

The flaws in the code are that in order to separate the shapes from the image in this way, we have reduced the surface of these shapes. Also, on the right side there are two very close-knit grains that are impossible to separate, while keeping the normal dimensions of the other grains. That is why they are recognized as one form.