Laboratory exercise: Shape parameters for binary images.

Area, moments and Fourier descriptors

In this exercise you will practice implementing the following shape parameters: area, moments, and Fourier descriptors. These parameters (=features) can be used to describe the shape of a binary object in a way that is convenient for classifying (recognizing) the object. To get a binary image from a grayscale image, you must assign each pixel to the background (zero) or to the object (one). In the simplest cases, this can be done by thresholding. If there is more than one object in the image we can assign a unique integer label to the pixels of each object. This will then allow us to compute the shape parameters for each object separately.

1. Area

Load the image *imbact*. Apply a threshold to the image in order to separate the pixels that belong to the background from those that represent the bacteria. Displaying the histogram of the image will make it easier to find an appropriate value T for the threshold. Display the grayscale image imbact, its histogram, and the binary image *bin_bact*.

```
load imbact; %load the image imbact
figure(1); subplot(2,2,1); imshow(imbact); %and display it
%manually find a threshold value T from the histogram of the image
[hist_bact,xx]=imhist(imbact); %compute the histogram
figure(1); subplot(2,2,2); stem(xx,hist_bact); %and display it
%make a binary image by thresholding
T=?; %put in the value for the threshold T here
bin_bact=imbact<T; %make a binary image
figure(1); subplot(2,2,3); imshow(bin_bact); %and display it
```

Compute the total area of the bacteria, i.e. the total number of pixels that constitute the bacteria.
Hint: all bacteria pixels have the value=1 and all background pixels have the value=0, so that summing up all pixel values in the binary image bin_bact will give you the total area of bacteria.

Now label pixels belonging to each bacterium with a unique label (**help bwlabel;**). Display the output image, which is in fact a segmented image and print the number of bacteria (=num in the code) on the screen.

```
[lab_bact,num]=bwlabel(bin_bact); %segment (label) the binary image
figure(2); subplot(2,2,1); imagesc(lab_bact);colormap(jet); axis image %display it
num %print on screen the number of objects
```

What is the result of the segmentation? Is the value of num correct?
Finally write a function area_obj **(function A=area_obj(lab_bact,num))** that takes as input parameters the segmented image lab_bact and the number of objects num in the

segmented image. The output parameter A should contain the area of each object in a column vector.
Use **area_obj** to compute the area of each bacterium in the binary image *bin_bact*. Display the areas and the area distribution estimated from your sample.

```
A_vect=area\_obj(lab_bact,num); %compute the area of each object
A_vect %print the area values on the screen
figure(2 ); subplot(2,1,2); hist(A _vect,20); %and the area distribution
```

Would you say that a ll the bacteria belong to the same family (if you consider the area to be a classifying param eter for a class of bacterium)?

2. Moments

Moments can be used to describe the shape of an object. For binary images a moment is defined as:

$$m_{pq} = \sum_{x,y \in \text{obj}} x^p y^q = \sum_{x,y} x^p y^q B(x,y)$$

The order of a moment is defined as $p+q$, which is the same as the order of the polynomial $x^p y^q$ in its definition. Note that the moment m_{00} is the area of the object, and $(m_{10}/m_{00}, m_{10}/m_{00})$ is the centroid (x_c, y_c) of the object.

$$\mu_{pq} = \sum_{x,y \in \text{obj}} (x-x_c)^p (y-y_c)^q = \sum_{x,y} (x-x_c)^p (y-y_c)^q B(x,y)$$

Write a function mom_obj ([F,m]=mom_obj(B)) that computes the following: $x_c, y_c, m_{00}, \mu_{11}, \mu_{20}, \mu_{02}$ of an object in a binary image $B(x,y)$. It is assumed that there is only one object in the image B . The features $\mu_{11}, \mu_{20}, \mu_{02}$ are put in a column vector F and x_c, y_c, are put in the column vector c.
Test with a simple object in two different orientations (for example a rectangle with 90° difference in orientation).

```
rect=zeros(21,21); %make a 7 by 3 rectangle image
rect(8:14,10:12)=ones(7,3);
figure(3);subplot(2,2,1); imshow(rect); %and display it
[F_rect,c_rect]=mom_obj(rect); %compute moments and centre
F_rect %print the moments [m0,0, 1,1, 2,0, and 0,2]
c_rect %print the centre [xm, ym]
```

Do the same for the rectangle image rotated by 90 degrees (rect=rect in the above code). Give comments on the values of the central moments 1,1, 2,0, and 0,2 for the two cases.
Also apply the function mom_obj to the binary images char_e and char_w (included in the exercise).

```
load char_e; %load image
load char_w; %
figure(3);subplot(2,2,3); imshow(char_e); %and display them
```

```
figure(3);subplot(2,2,4); imshow(char_w);
[F_e,c_e]=mom_obj(char_e); %compute moments and centre
[F_w,c_w]=mom_obj(char_w); %
[F_e, F_w] %and print them on screen
[c_e, c_w]
```

Could the features in F_e respective in F_w be used to distinguish between the two characters? Can we use c_e, c_w instead?

3. Fourier descriptors

Fourier descriptors (Fd:s) can be used to describe the boundary of an object. This description offers a possibility to smooth the boundary (compressed description) and to make the description invariant to translation, rotation, and scale. Run the following code:

```
load char_e; %load character E
figure(4); subplot(2,2,1); imshow(char_e); %and display it
[x y]=perim_sort(char_e); %extract and sort the boundary coordinates
f=x+i*y; %make a complex 1D signal out of the coordinates
N=length(f); %number of boundary points (N=262)
figure(4); subplot(2,2,2); plot(imag(f),real(f)); %plot the boundary
F=fft(f,N); %Fdescriptors of the boundary
fi=ifft(F,N); %go back to boundary points x+i*y
figure(4); subplot(2,2,3); plot(imag(fi),real(fi)); % and plot the boundary
```

The high order Fd:s describe the sharp variations in the boundary (high frequencies) and the low order Fd:s describe the smooth variations (low frequencies). Therefore, if we suppress the high order Fd:s and then reconstruct the boundary from the low order Fd:s the boundary should look smoother. Run the following code:

```
CC=zeros(N,1); %start with a zero filter
% the Fd:s should be put in in pair, i.e. F(k) and F(-k) is a pair
% in the Brick-wall Fourier domain filter
CC=zeros(N,1); %start with a zero filter
CC(mod([0:29],N)+1)=ones(30,1); %add in ones in pair; F(k)
CC(mod([-29:-1],N)+1)=ones(29,1); %F(-k)
FC=F.*CC; % filter
f1=ifft(FC,N); % reconstruct the boundary from the low order Fd:s
figure(4); subplot(2,2,4); plot(imag(f1),real(f1)); %reconstructed boundary
```

*Explain how the filtering is done in the frequency domain (FC=F.*CC in the code) by writing down the filter CC and the Fd:s (=F) and manually do the multiplication.*
Try smoothing the boundary even more (by changing the filter CC in the code above). Display your results. For example use the first 5, 8, and 12 Fd:s.
Filter out the first Fd only (i.e. F(0)) and then reconstruct the boundary. Explain your result.