Assignment 4

Segmentation

Name: Gowri Kurthkoti Sridhara Rao

Introduction

In digital image processing and computer vision, image segmentation is the process of partitioning a digital image into multiple segments. The goal of segmentation is to simplify and/or change the representation of an image into something that is more meaningful and easier to analyze. Image segmentation is typically used to locate objects and boundaries (lines, curves, etc.) in images. More precisely, image segmentation is the process of assigning a label to every pixel in an image such that pixels with the same label share certain characteristics.

There are two major types of image segmentation,

1. Semantic image segmentation
2. Instance image segmentation

There are several methods of carrying out this process of image segmentation:

1. Region based image segmentation
2. Edge detection-based image segmentation
3. Image segmentation based on clustering

In this project, region-based segmentation is used. The region-based segmentation method looks for similarities between adjacent pixels. That is, pixels that possess similar attributes are grouped into unique regions. As with all segmentation techniques, using gray-level intensity is the most common means of assigning similarity, but other possibilities exist, such as variance, color, and multispectral features. Region-growing techniques cluster the pixels that represent homogeneous areas in an image. Regions are grown by grouping adjacent pixels whose properties, such as intensity, differ by less than some specified amount. Each grown region is assigned a unique integer label in the output image. This class of algorithms tends to work well for difficult imagery, as it is adaptive and less susceptible to the effects of partial occlusion, adjacency, noise, and ambiguous boundaries.

In its simplest form, the region-grown operator performs connected-components analysis on gray-scale pixel values. At each pixel in an image, neighboring pixels are compared to a reference pixel value for the region. If the difference is less than or equal to the Difference Threshold, the neighboring pixel is added to the current region.

In this project, 8 neighboring pixels are considered to decide the region of the pixel. If more than 5 belong to the background, the pixel is considered to belong to the background. This helps us ignore the noisy small regions which don't belong to the image completely. Among the given images, one image had such small regions which affected the centroid calculation of the main region. 8-connectedness implementation in the code helped in ignoring such regions and identifying the centroid and other moments correctly.

Segmentation – Region growing method

Steps carried out to perform region growing, calculating moments, centroids and principal axes are:

1. The image is divided into regions, the background is numbered region 1 and object is numbered region 2 based on the intensity of the pixel.(background =0, object = 255)
2. 8-connected check is performed on each pixel to identify if any noisy regions are present.
3. Moments are calculated for region 2 since we are interested only in its moments.
4. Using the moments, centroid, and the orientation of the region is computed
5. Plotting centroid and principal axes on the image

Code

1. <u>Region growing</u>

```
img = imread("dif00002.pgm"); %Input image
%% Region-growing
region=zeros(size(img)); %Initializing a matrix to hold region values
regions(1) = img(1,1); %List to hold pixel intensity value for each region
and initializing it to first pixel
current_region = 1; %Initializing the current region
for i=1:size(img,1)
    for j=1:size(img,2)
        found=0; %reset Classiffication flag each iteration
        if(img(i,j) == regions(current_region)) %If the pixel belongs to the
current region
            region(i,j) = current_region; %Set the value of current region
to the  region pixel
            found=1; %Set classified flag
        else %If pixel does not belong to the region,
            for k=1:size(regions,2) % Check for intensity value that matches
the pixel in the
                if(regions(k)==img(i,j)) %regions list. If found, set the
current region to k
                    current_region=k;
                    found=1; %Set classified flag
                end
            end
            if(found==0) %If the regions list does not have that value
                current_region = current_region+1; %Increment the current
region value
                regions(current_region)=img(i,j); %Add the intensity value
to the list
            end
            region(i,j) = current_region; %Set the value of current region
to the  region pixel
        end
    end
end
```

2. <u>8-connected check</u>

```
%% 8-connected
neigh = [1 1 1;1 0 1;1 1 1]; %8 surrounding neighbors
for x=1:size(region,1)-1 %Iterate through the region matrix
    for y=1:size(region,2)-1
        if (region(x,y)==2) %If the pixel belongs to object
```

```
            res    =    ((region(x-1:x+1,y-1:y+1).*neigh)==1);%Observe    8
neighbours of current pixel check how many are equal to 1
            if(sum(res)>=5) %If more than or equal to 5 belong to region 1
or background,
                region(x,y)=1; %Set the region to 1(background)
            end
        end
    end
end
```

3. <u>Moment calculation</u>

```
%% Calculating moments
clear M00; clear M01; clear M10; clear M11; clear M02; clear M20;
M00 = 0;M01 = 0;M10 = 0;M11 = 0;M02=0;M20=0;
fimg = 1; %Function f(x,y)=1 for the object
j=1;
for x= 1:size(img,1) %Iterate through the region matrix
for y = 1:size(img,2)
if(region(x,y)==2) %If the region value is 2
x_coords(1,j)=x; %Copy the x value
y_coords(1,j)=y; %copy the y value
j=j+1;
M00 = M00 + (x^0)*(y^0)*fimg; %Calculate moments
M01 = M01 + (x^0)*(y^1)*fimg;
M10 = M10 + (x^1)*(y^0)*fimg;
M11 = M11 + (x^1)*(y^1)*fimg;
M02 = M02 + (x^0)*(y^2)*fimg;
M20 = M20 + (x^2)*(y^0)*fimg;
end
end
end
```

4. <u>Centroid and orientation computation</u>

```
%% Computations

%Centroid
y_bar = round(M10/M00);
x_bar = round(M01/M00);

%Orientation
U11 = M11 - ((M10*M01)/M00);
U20 = M20 - ((M10^2)/M00);
U02 = M02 - ((M01^2)/M00);
x_theta = 2*U11/sqrt(4*(U11^2)+(U20-U02)^2);
y_theta = (U20-U02)/sqrt(4*(U11^2)+(U20-U02)^2);
thetad = 0.5*atan2d(x_theta,y_theta);
thetar = pi*thetad/180;
R = [ cos(thetar)    sin(thetar)
     -sin(thetar)    cos(thetar)];

%Maximum length
max_len = x_coords(length(x_coords))-(x_coords(1));


x = [min(y_coords):max(y_coords)]; %Length of vertical line
```

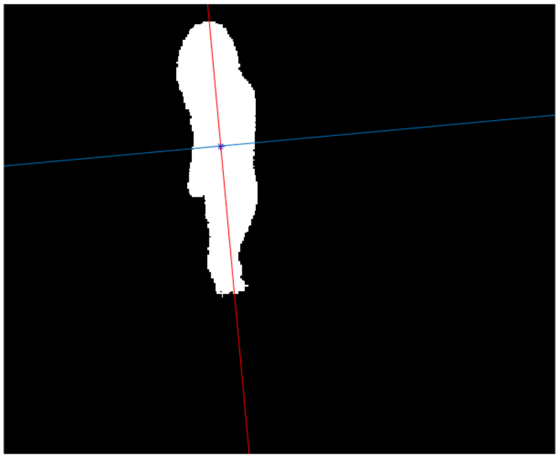
```
y_len = [min(x_coords):max(x_coords)]; %Length of horizontal line
```

5. Plotting

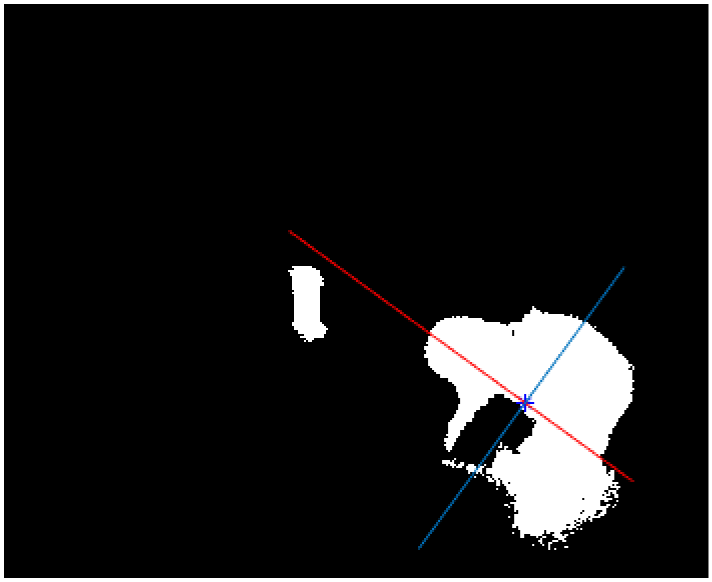
```
%% Plotting
m=R(1,1)/R(1,2); %Slope = cos(theta)/sin(theta)
c = y_bar-m*x_bar; %Intercept of the line passing through centroid
% Vertical Principal axis
verti = m.*x + c; %Equation of vertical line

%Horizontal Principal axis
thetad = thetad+90; %Horizontal axis is perpendicular to vertical line
thetar = pi*thetad/180;
m1 = sin(thetar)/cos(thetar); %Slope in this case is sin(theta)/cos(theta)
% m1 = 1/(m); % Slope of the like that is perpendicular to vertical line

c1 = x_bar-m1*y_bar; %Intercept of the line passing through centroid
hori = (m1.*y_len +c1); %Equation of horizontal line

figure();
imshow(img); %Show image
hold on
grid on
plot(x_bar, y_bar, 'b*'); %Plotting centroid
plot(x, verti,'r-'); %Plotting vertical line
plot(hori,y_len); %Plotting horizontal line
hold off
grid off
```

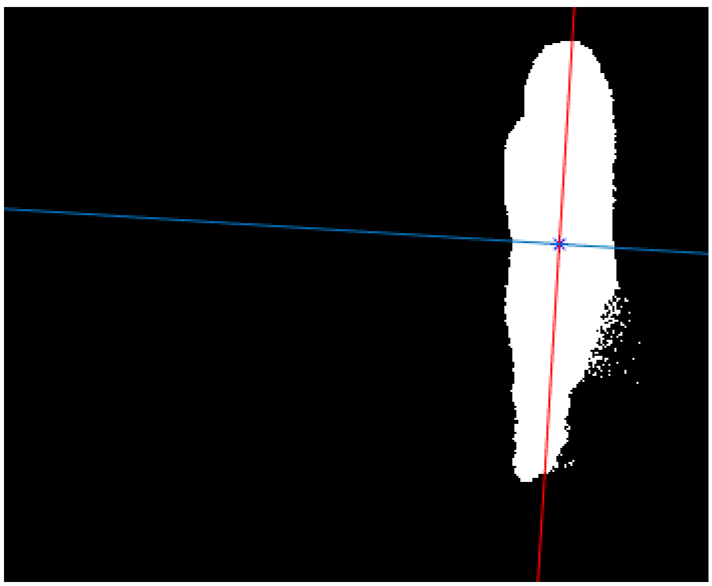
Results

A set of images were provided, and each of them were images that were already thresholded. The task that was assigned was to perform region growing and identifying the region centroid and orientation using moments. Below are the result images with centroid and principal axes marked.



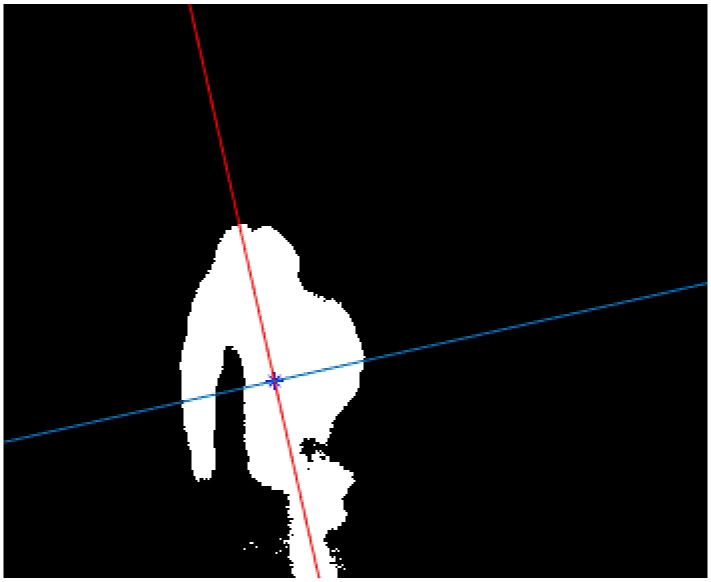
Segmentation – Region growing method

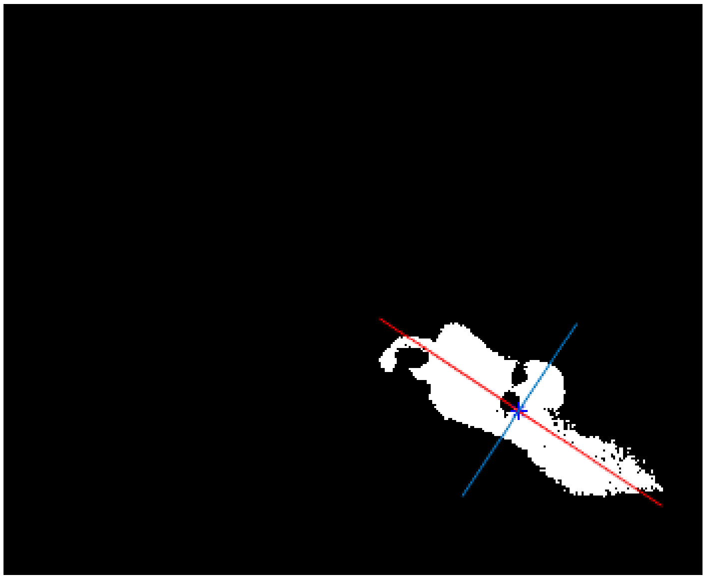




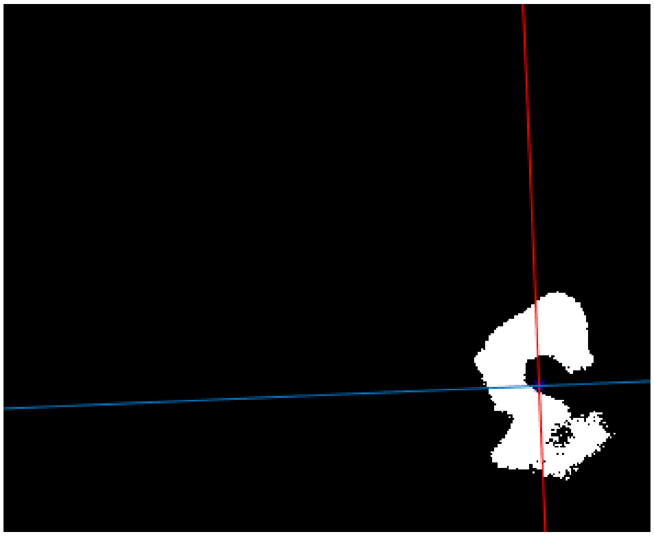
Due to 8-connected approach, the smaller regions that do not belong to the image completely are ignored while calculating the moments, centroid and orientation.

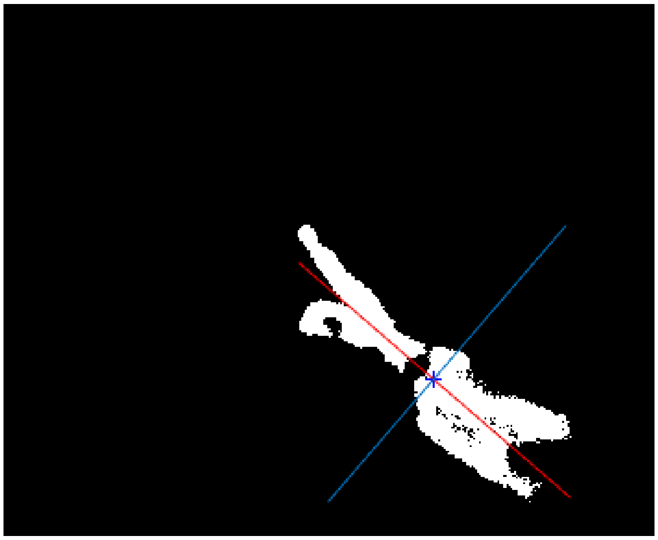
Segmentation – Region growing method

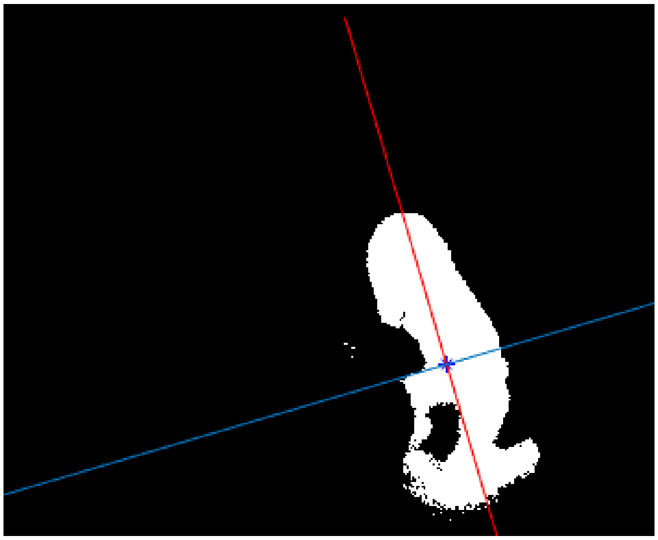




Segmentation – Region growing method







Discussion

The above set of result images show that the region is detected properly and the not connected parts were ignored while computing the moments. Due to this process, in image 2, another region which was not connected to the main region was ignored and its moments were not calculated. The algorithm implemented was not able to identify that both the regions are different and hence moments have to be computed separately. This problem can be momentarily solved by using in-built function region props. This function provides information regarding all the regions in the image. Whereas, it is not the case with the code implemented in this project. This is found to be the limitation of the project and hence the future work will revolve around this even after the project submission.

Conclusion

Segmentation is performed on the given set of images using the region growing algorithms. Region growing using pixel intensity value as the similarity check component is performed backed by the 8-connected decision making, the pixels are divided into regions and their properties like centroid and principal axes are calculated. Results and the code are presented in the report.