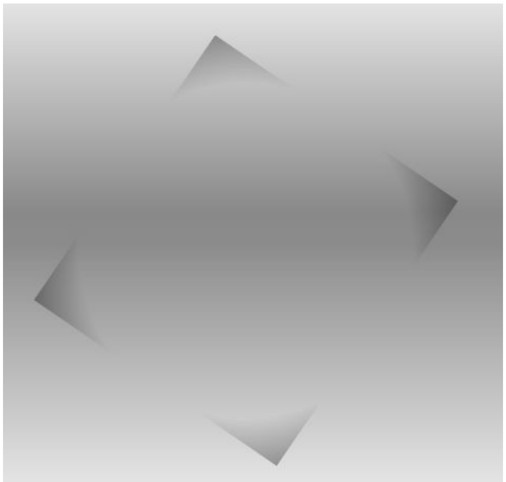
Lab 3: Hough Transform(s) - Voting Spaces

The goal of this lab is to implement and test different implementations of the Hough Transform in Matlab.

I. Modeling of line segments



We will try to identify the line segments that form the edges of the rectangle that can be guessed in the image opposite. We will model the lines by their equation in polar coordinates. To do this:

- Produce a binary image of the edges using a gradient estimator and a threshold. **Give this binary image.**

- Construct and visualize the corresponding voting space. For example, we will take 628 value intervals for the angular parameter, and all possible integer values of the radial parameter. The display of the voting space can be done with the following instructions:

```
imagesc(log(V+1));
colormap(gray(256));
```

where V is the voting matrix. **Give the image of this voting matrix.**

II. Recognition of line segments

Identify the coordinates of the k points receiving the largest number of votes (when k varies from 1 to 20). To each of these points of coordinates (rho, theta) corresponds a line in the image space. Find for each line the coordinates of the points (x1,y1) and (x2,y2) at 1 in the binarized contour image, the farthest apart, and whose coordinates verify the equation:

abs(x.cos(theta) + y.sin(theta) - rho) < epsilum

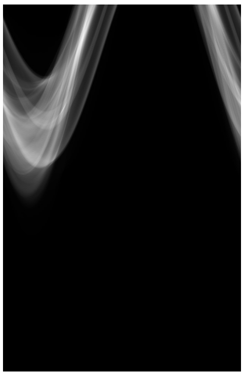
where epsilum is a positive value close to zero.

Reconstruct the original shape by drawing the corresponding lines using the instructions :

```
hold on;
plot([y1 y2],[x1 x2],'-w');
```

Observe the results obtained when k varies from 1 to 20. **Give the images corresponding to k=1, k=4, k=10 and k=20.**

III. Back projection of the Radon transform



This matrix is the result of a CT scan of a part of a human body. By applying the Radon back-projection formula, **give a representation of the corresponding internal organs.**

The horizontal axis represents the theta angle for radian values ranging from 0 to 6.27 from left to right. One pixel represents a step of 0.01 radian. The vertical axis represents rho for values ranging from 1 to 968.

The reconstructed image is required to have a resolution of 468 rows by 500 columns of pixels.

IV Generalized Hough Transform

IV.1 Subroutines

In a first step, you are asked to prepare a set of functions that will be used later.

IV.1.a Function "contour"

Copy the function contour.m that you will obtain by clicking on the link. This function allows you to retrieve the list of points located on the contour of a shape contained in a binary image. The parameter of the function is a 2D matrix containing only 0's and 1's. **Determine the format and the meaning of the values produced in the 2 result variables "listcontour" and "nbpoints".**

IV.1.b Function "beta"

Copy the function beta.m that you will obtain by clicking on the link. This function allows you to retrieve the value of the angle of the perpendicular to the tangent towards the interior of the shape at a given point of the contour. **Determine the expected input values for this function and their role.**

IV.1.c Function "barycentre" (barycenter)

Write the function "barycentre.m" which computes and returns the coordinates of the center of gravity of a shape contained in a binary image passed as a parameter :

```
function [xo,yo]=barycentre(img)
```

IV.1.d Function "alpha"

Copy the function alpha.m that you will obtain by clicking on the link. This function allows to get the value of the angle between the line connecting a point of the contour and the center of gravity with the x-axis. **Determine the expected input values for this function and their role.**

IV.1.e Function "distance"

Write the function "distance.m" which calculates and returns the Euclidean distance between two points whose coordinates are passed in parameter:

```
function d=distance(x1,y1,x2,y2)
```

IV.2 Modeling

We are going to produce the "model" of the form opposite, in the sense of the generalized Hough transform. To do this, **complete in the program below the part marked by "..." according to the comment which precedes it:**

```
img=imread('TP07I02.bmp');
img=img(:,:,1)>0;

% Recovering information about the contour
[C, N]=contour(img);
% Initialize the model :
% row = beta value * 100
% column = order number of the couple (alpha, distance)
% third dimension: for 1 = alpha, for 2 = distance
H=zeros(round(100*2*pi),N,2);

% Calculation of the coordinates of the barycenter
...

% Browsing the points on the contour
for i=1:N

    % calculation of b=beta for the ith point of the contour
    ...

    % search for the first "empty" column in the table on the
line beta
    k=1;
    while H(b+1,k,2)~=0
        k=k+1;
    end

    % record of the value of alpha and the distance on the b+1
row and the kth column of H
    ...
    ...
end
```

IV.3 Recognition

We will finally apply the recognition process using the model learned in IV.2. By modifying the previous program, **produce the voting space corresponding to each of the 4 shapes given below**. An example of each voting spaces is given. To display the image of this voting space V, we used the instructions :

```
imagesc(log(V+1));
colormap(gray(256));
```

Check that the image of the model is indeed the one which produces the highest vote in the spaces of votes thus produced. **Give in each case, the value of this vote.**

Proceed in the same way by choosing in turn the 2nd, 3rd and 4th image as a template.

Image binaire	Voting space