Faculty of Engineering
Cairo University

Computer Vision

Task II

Team 7

Name	ID
Ahmed Hassan	9202076
Dina Khalid	9202529
Abdelrahman Yasser	9202808
Mohammed Nasser	9203348
Mahmoud Rabea	9203396

-

Table of Contents

-

Hough Transform

Hough Transform is considered as a feature extraction technique that is widely used in various fields including: image processing, computer vision and much more.

It's used to locate instances of objects using certain shapes – usually lines, circles and ellipses – this is done by a voting procedure in the parameter space.

In the early stages of the Hough transform, it was commonly used associated with only the identification of lines in every image but now, it has extended to include more shapes such as ellipses and circles.

The main algorithm behind detecting lines using Hough transform is as follows:

1- Applying canny edge detection technique to the image.
2- Map the edge points to the parameter space "Hough space" and store in an accumulator.
3- Interpret the accumulator to lines of infinite length and it's usually done by thresholding.
4- Conversion of the infinite lines into finite lines.

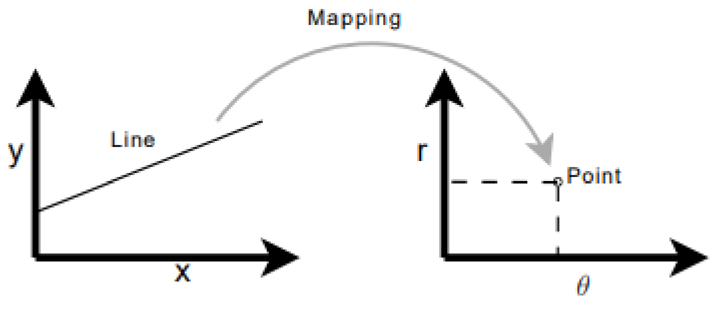
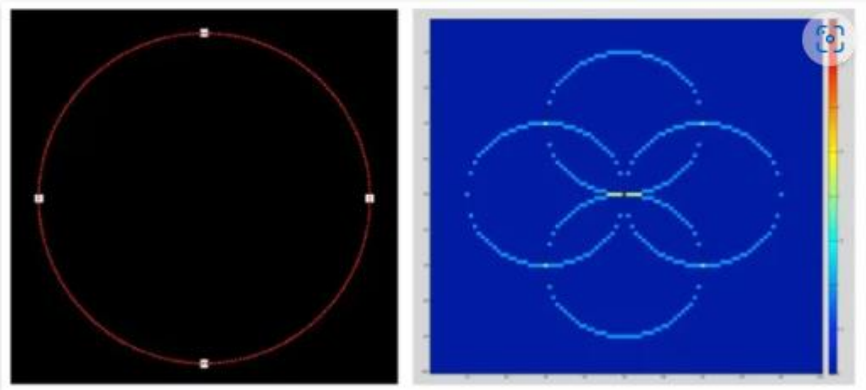


Figure 1: *Mapping of one unique line to the Hough space.*

The main algorithm behind detecting circles using Hough transform is as follows:

1- Initialize the accumulator to all zeros.
2- Convert the image to gray scaled image.
3- Applying edge detection using any technique –preferably canny edge detection–.
4- Vote on all possible circles in the accumulator.
5- Having two thresholds: R_max and R_min indicating the local maximum and minimum of the voted circles of the accumulator in the Hough space.
6- Finally, the maximum voted circle found in the accumulator shall indicate the circle.



The algorithm for ellipses takes two different points belonging to the ellipse and assumes that they are the main axis. Then it loops over all the other points and determines how much an ellipse passes through them. A good match corresponds to high accumulator values. The algorithm has five parameters to estimate: the center coordinates, the major and minor axes, and the orientation of the ellipse.

In the case of ellipse detection, the accumulator is a five-dimensional array that stores the votes for each possible combination of center coordinates, major and minor axes, and orientation of the ellipse.

The higher the accumulator value, the more likely that there is an ellipse with those parameters in the image. The accuracy of the Hough transform depends on the number of accumulator cells you have. The more cells you have along a particular axis, the more accurate the transform would be, The voting procedure helps to improve the feature detection accuracy and overcome cases of missing or noisy feature parts.
The Hough transform parameters for ellipse detection are:

- Image: (M, N) Input image with nonzero values representing edges from canny edge detection.
- Threshold: int, optional Accumulator threshold value.

- Accuracy: double, optional Accuracy of the result. Increase to get better precision.
- min_size of ellipse: int, optional Minimum size of searched ellipse.
- max_size of ellipse: int, optional Maximum size of searched ellipse.

These parameters are used by the hough_ellipse function, which performs an elliptical Hough transform and returns a list of best-detected ellipses.

Some of the advantages and disadvantages of the Hough transform technique are:

Advantages:

- It is robust to partial or slightly deformed shapes (i.e., robust to recognition under occlusion).
- It is robust to the presence of additional structures in the image.
- It is tolerant to noise.
- It can find multiple occurrences of a shape during the same processing pass.

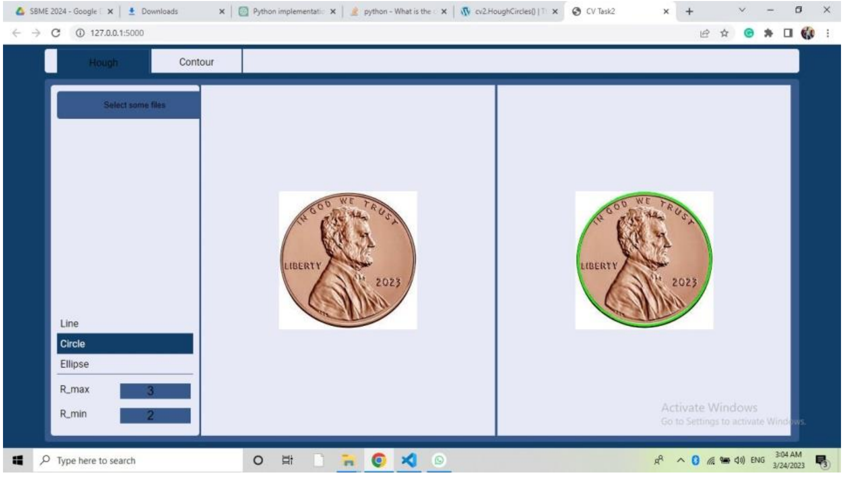
Disadvantages:

- It requires a large amount of memory and computation time for high-dimensional parameter spaces.
- It may produce spurious peaks or miss true peaks due to quantization effects or overlapping shapes.
- It may require a threshold or a peak detection method to identify the relevant parameters.

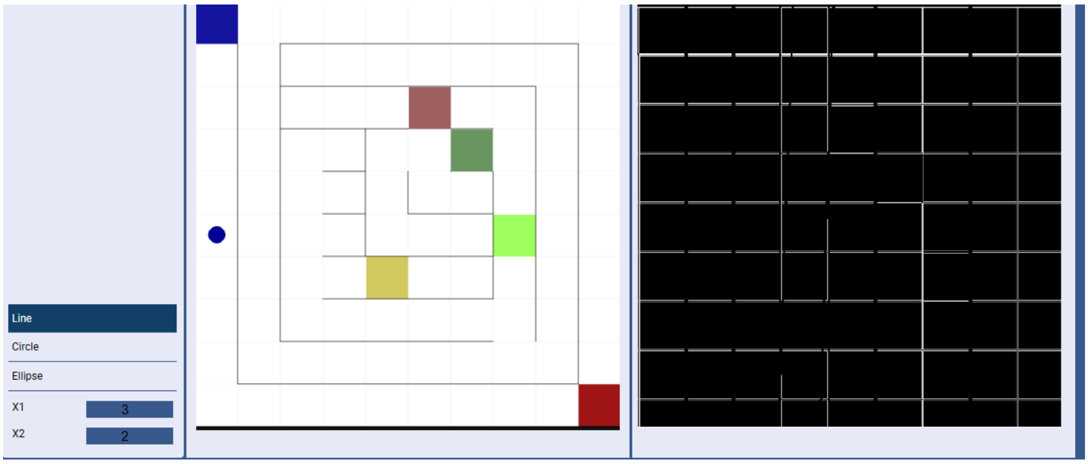
It may not be able to detect complex or irregular shapes that do not have a simple parametric representation

Output samples

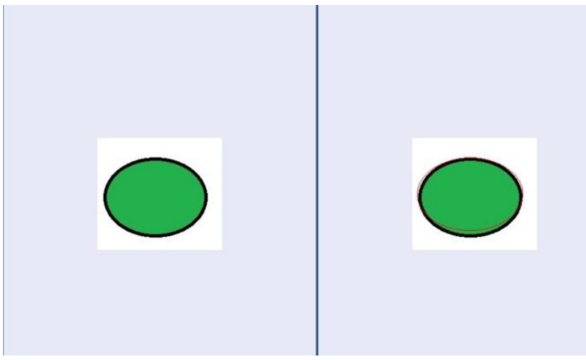
Output sample for circle identification:

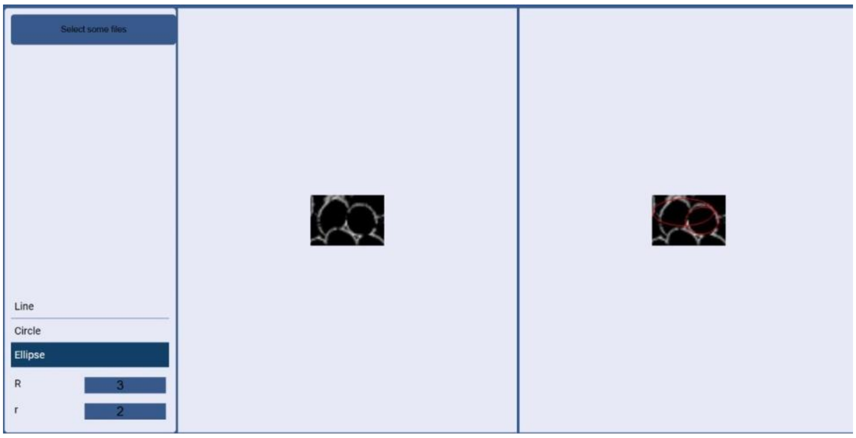


Output sample for line identification:



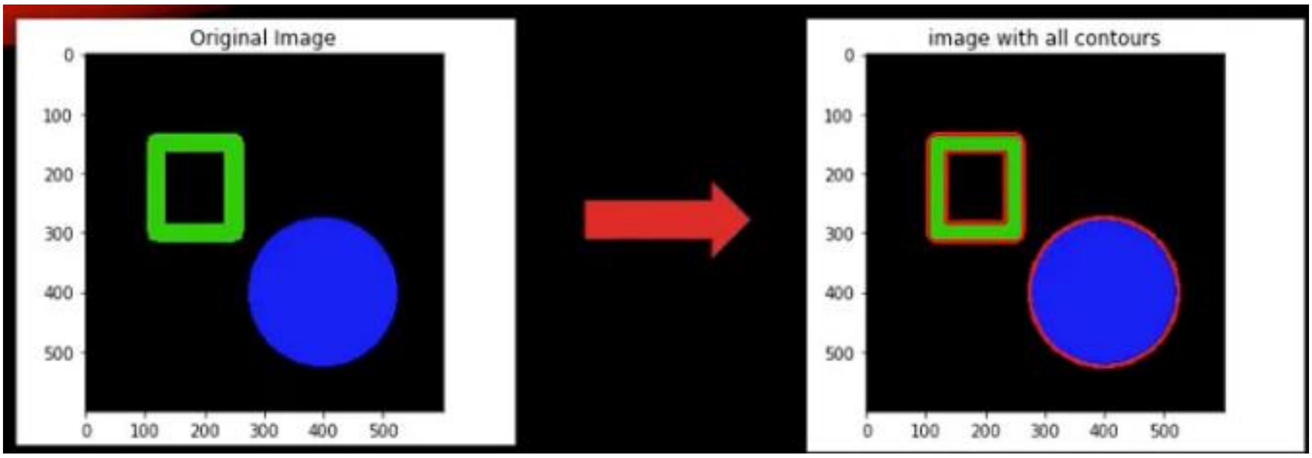
Output sample for ellipse identification:





Active Contour

Contouring an image means to identify all the object's outlines which may help us to claim the shape of this particular object.



It's widely needed in image processing in order to define smooth shapes in images and locate uneven shapes in an image. In medical applications, active contours are commonly used to examine slice of the brain obtained using CT scan.

Main active contour models are:

1- Snake model:

$$\mathcal{E} = \int (\alpha E_{cont} + \beta E_{curv} + \gamma E_{image})\, ds$$

2- Gradient vector flow model:

$$E_{\mathrm{GVF}} = \iint \mu(u_x^2 + u_y^2 + v_x^2 + v_y^2) + |\nabla f|^2 |\mathbf{v} - \nabla f|^2 \, dx\, dy$$

3- Balloon model: $F_{\text{inflation}} = k_1 \vec{n}(s)$

4- Geometric active e contour model: $\frac{\partial \Phi}{\partial t} = |\nabla \Phi| \operatorname{div}\left(g(I)\frac{\nabla \Phi}{|\nabla \Phi|}\right) + cg(I)|\nabla \Phi|$

Our concern is Snake model which depend on (alpha/beta/gamma) which represents three different energy settings:

$$\mathcal{E} = \int (\alpha E_{cont} + \beta E_{curv} + \gamma E_{image})\, ds$$

The Snake Greedy algorithm works as follows:

1- Create pre-defined points based on snake spacing.

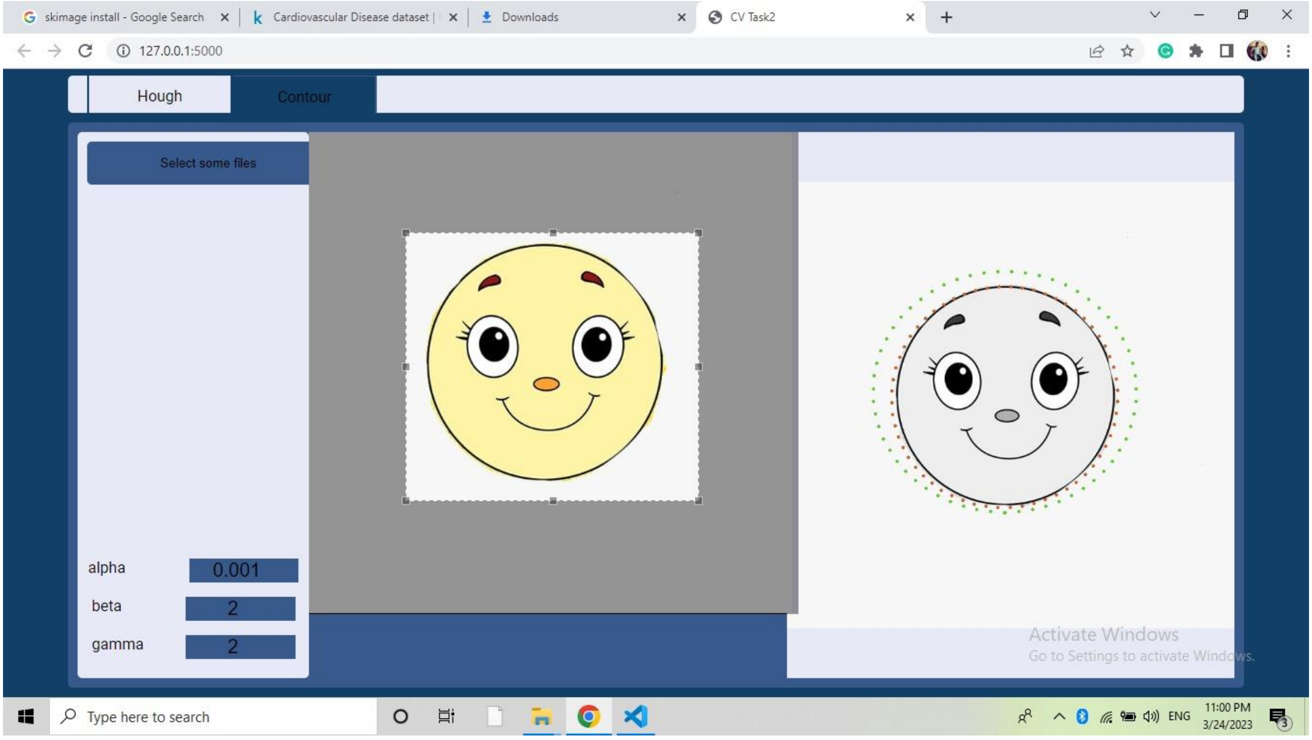
2- Points move to the lowest energy in the local neighborhood that is identified by the energy function.

3- The process keeps going until the snake stops.

Note that choosing a proper value for each parameter depend on the feature you want to extract so:

- Set alpha high if deceptive image gradient is located.
- Set beta high for smooth edged feature and low for sharp edges.
- Set gamma high when you find contrast between the background and feature is low.

Output samples



Chain code:

```
Chain code: [3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3,
 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 3, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 2, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1,
 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 1, 7, 7, 7, 7, 7, 7,
 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7,
 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 7, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 6, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5,
 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5, 5]
Perimeter: 1131.142406463623
Area: 101681.5
```