Computer Vision

Task (2)

Name	Section	Bench Number
Amr Mohamed	2	7
Adham Mohamed	1	9
Belal Mohamed	1	23
Mahmoud Emad	2	28
Ezzat Hegazy	1	56
Mina Azer	1	58

Submitted To:

Prof. Dr. Ahmed Badawi

TA. Peter Emad

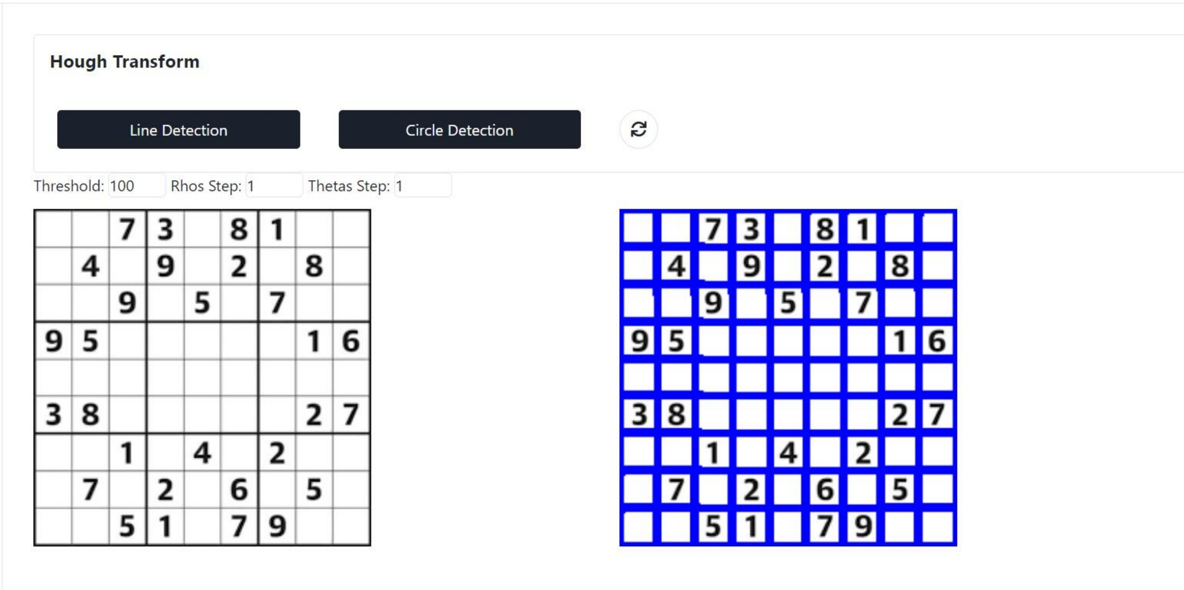
Systems and Biomedical Engineering Dept.

Cairo University

The Hough Transform: is a popular technique for detecting straight lines in an image. It works by converting each pixel in an image to a point in parameter space, where each point corresponds to a possible line in the image. The algorithm then identifies the points in parameter space that correspond to lines in the image.

1. **Line detection using Hough Transform**:
   The steps for line detection using the Hough Transform algorithm are as follows:
   i. Convert the image to grayscale.
   ii. Apply a Canny edge detection algorithm to obtain the edges in the image.
   iii. Calculate the two arrays of all possible Rohs and thetas by the needed step given by the user.
   iv. Create an accumulator array in Hough space, where each point represents a line in image space.
   v. For each edge pixel in the image, calculate the Roh and for each theta from that pixel to every possible line in image space.
   vi. Increment the corresponding point in the accumulator array for each possible line that intersects the edge pixel.

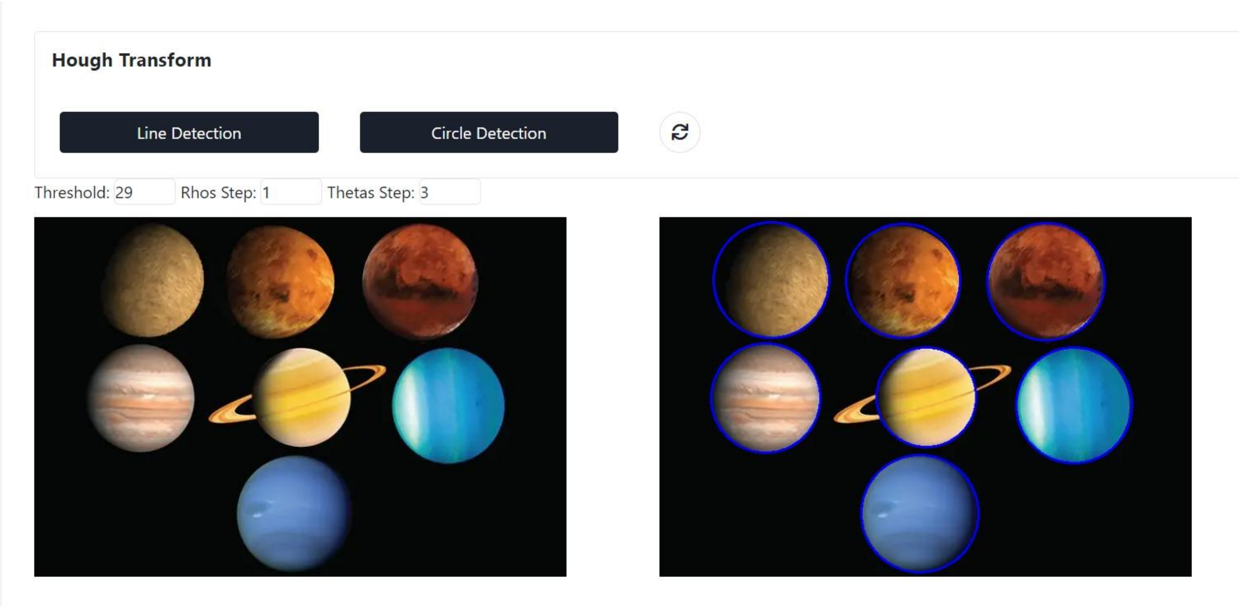
   vii. Find the local maxima in the accumulator array, which represent the lines with the most edge pixels that intersect them.
   viii. Convert the lines in Hough space back to image space and draw them on the original image.



2. **Circle detection using Hough Transform**: The steps for circle detection using the Hough Transform algorithm are as follows:
   I. Convert the image to grayscale.
   II. Apply a Canny edge detection algorithm to obtain the edges in the image.
   III. Create an accumulator array in Hough space, where each point represents a circle in image space.
   IV. Calculate the two arrays of all possible Rohs and thetas by the needed step given by the user.
   V. For each edge pixel in the image, calculate the each possible center of the circle for each rhos and thetas that pixel to every possible circle in image space.
   VI. Increment the corresponding point in the accumulator array for each possible circle that intersects the edge pixel.

VII. Find the local maxima in the accumulator array, which represent the circles with the most edge pixels that intersect them.

VIII. Convert the circles in Hough space back to image space and draw them on the original image.



The Active Contour (Snake): is a computer vision technique used for image segmentation. It is a type of deformable model that can be used to extract the boundary of an object in an image.

1. Initialize the snake:

   I. Define an initial contour or snake shape.

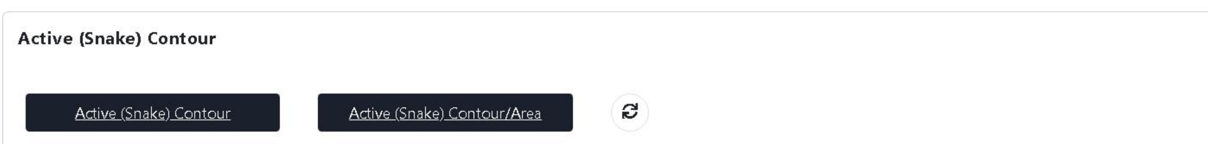
   II. Set the parameters for the snake model, such as the weight of internal and external energy terms, and the number of iterations.

2. Compute external energy:

   I. Calculate the gradient of an image or a feature map to obtain edge information.
   II. Compute external energy by using this edge information to attract the snake towards object boundaries.

3. Compute internal energy:
   I. Define a set of internal energy functions that control the smoothness and shape of the snake.
   II. Compute internal energy by evaluating these functions at each point on the snake.
4. Update snake position:
   I. Move each point on the snake in a direction that minimizes total energy (internal + external).
5. Resample the snake:
   I. Create a uniform spacing between each point in the snake
6. Repeat steps 2-5 until convergence:
   I. Iterate until the snake converges to an object boundary or reaches a maximum number of iterations.

7. Output final contour: Return the final position of each point on the snake as a contour that outlines an object in an image or feature map.



8. We compute the perimeter and the area inside these contours using ear clapping method where we divide the contour to mesh of triangles.

The Chain code representation: is a technique used to represent the boundary of an object in an image. In active contour algorithm, it consists of a sequence of directions that describe the path of the boundary. Each direction is represented by a number from 0 to 7, where 0 represents a horizontal direction and the other numbers represent diagonal directions.

1. Start at a point on the contour.

2. Choose a direction to move in (usually clockwise or counterclockwise).

3. Move one pixel in the chosen direction.

4. Check the 8 neighboring pixels to see which one is part of the contour.

5. Move to that pixel and mark it as visited.

6. Repeat steps 4 and 5 until you return to the starting point.
7. Record the direction of each movement as a code (0-7) and store it in a list.
8. Repeat steps 2-7 for all remaining points on the contour.
9. Use the list of codes to reconstruct the contour by starting at any point and following the directions in order, connecting each pixel to its neighbor according to the code.
10. The resulting chain of connected pixels represents the active contour of the object in question.