CSE 573: Introduction to Computer Vision and Image Processing
(Fall 2018)
Instructor: Junsong Yuan

Project 3

December 3, 2018

Report By:
Siddheswar Chandrasekhar

Objective

The objective is to perform three independent tasks: Morphology Image Processing, Image Segmentation and Point Detection, and Hough Transform

Task 1: Morphology Image Processing

Morphological image processing is a collection of non-linear operations related to the shape or morphology of features in an image. Morphological operations rely only on the relative ordering of pixel values, not on their numerical values, and therefore are especially suited to the processing of binary images. Morphological operations can also be applied to greyscale images such that their light transfer functions are unknown and therefore their absolute pixel values are of no or minor interest.

Morphological techniques probe an image with a small shape or template called a structuring element. The structuring element is positioned at all possible locations in the image and it is compared with the corresponding neighborhood of pixels. Some operations test whether the element "fits" within the neighborhood, while others test whether it "hits" or intersects the neighborhood. [1]

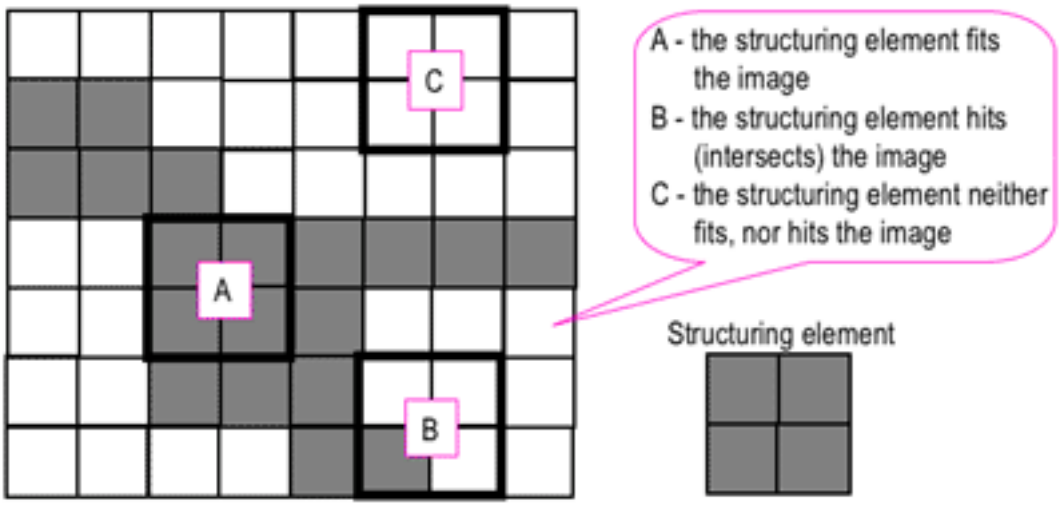

Fig 1.1 Morphology Image Processing

A morphological operation on a binary image creates a new binary image in which the pixel has a non-zero value only if the test is successful at that location in the input image.
The structuring element is a small binary image, i.e. a small matrix of pixels, each with a value of zero or one:

- The matrix dimensions specify the size of the structuring element.
- The pattern of ones and zeros specifies the shape of the structuring element.
- An origin of the structuring element is usually one of its pixels, although generally the origin can be outside the structuring element.

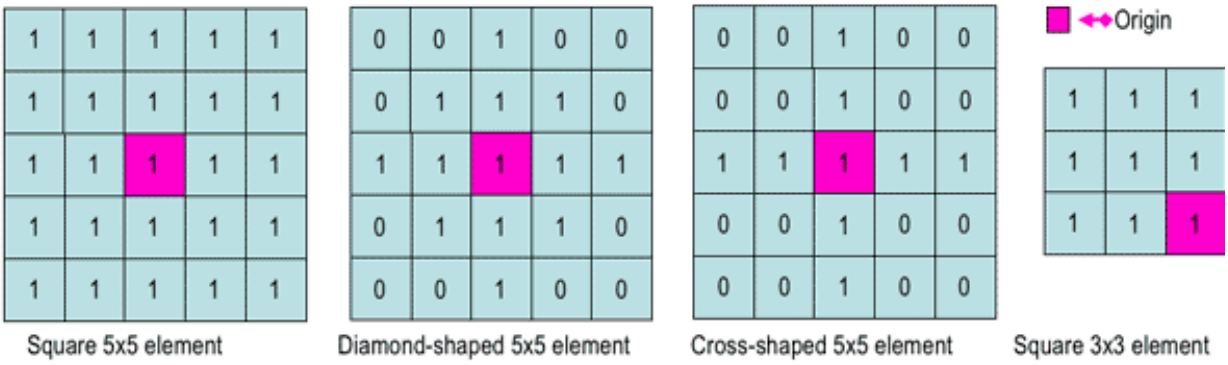

Fig 1.2 Examples of simple structuring elements

Morphological Operators

Primary morphological operations are Dilation and Erosion. More complicated morphological operators such as Opening and Closing can be designed by means of combining erosions and dilations.

- Opening generally smoothens the contour of an image and eliminates protrusions.
- Closing smoothens sections of contours, but it generally fuses breaks, holes and gaps.

Dilation of A and B is defined as:

$$A \oplus B = \left\{ z \mid \left[\left(\hat{B} \right)_z \cap A \right] \neq \varnothing \right\}$$

- The effect of dilation with 3 x 3 mask is to add a single layer of pixels to the outer edge of an object and to decrease by a single layer of pixels to the holes in the object.
- A 5 x 5 mask will add two layers of pixels which is equivalent to applying a 3 x 3 mask twice.
- The main application of dilation is to remove small holes from the interior of an object.

Erosion of A and B is defined as:

$$A \ominus B = \left\{ z \mid \left(B \right)_z \subseteq A \right\}$$

- The effect of erosion with 3 x 3 mask is to strip a single layer of pixels to the outer edge of an object and to increase by a single layer of pixels to the holes in the object.
- A 5 x 5 mask will strip off two layers of pixels which is equivalent to applying a 3 x 3 mask twice.
- The main application of dilation is to remove small noise artifacts from an image.

A compound operation is when two or more morphological operations are performed in succession. A common example is Opening which is a combination of erosion followed by a dilation:

$$A \circ B = \left(A \ominus B \right) \oplus B$$

- Opening is often performed to clear an image of noise whilst retaining the original object size.
- The opening operation tend to flatten the sharp peninsular projections on the object.

Closing is the complementary operation of opening, defined as dilation followed by erosion:

$$A \bullet B = \left(A \oplus B \right) ! \ \ B$$

Our task is to use these two morphological algorithms to remove noise from our image and compare the results of the two.

We perform a combination of Opening followed by Closing as one operation, and Closing followed by Opening as another.

Opening + Closing Closing + Opening

Fig 1.3 Morphology Algorithm output

As we can see, the output of the two operations is similar.

This is because the combination of opening and closing is performed to remove all noises from the image. But the two complex tasks (i.e. opening and closing) is independent of each other, and hence it doesn't matter which operation we perform first, as long as we perform both the output will be the same.

We then perform morphological image processing operations to extract the boundaries of the noise-free image.

The boundary of a set A can be defined as:

$$\beta(A) = A - (A \,!\; B)$$

The output is as follows:

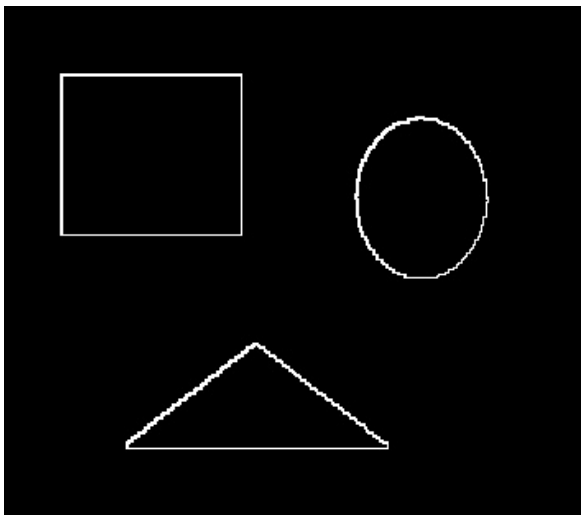

Fig 1.4 Boundary of Opening + Closing

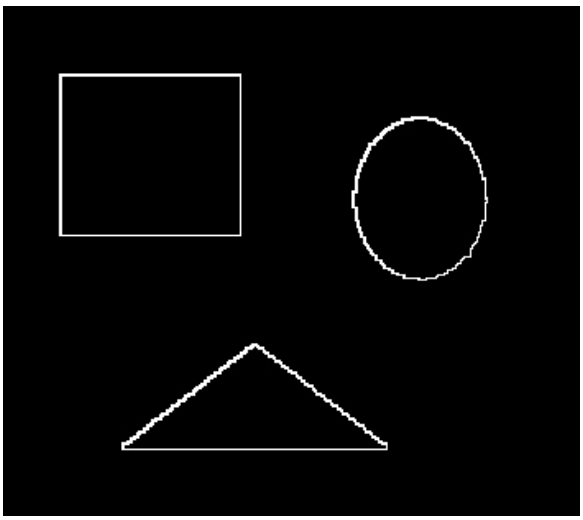

Fig 1.5 Boundary of Closing + Opening

As expected, the boundaries of the two images are also similar.

Task 2: Image Segmentation and Point Detection

We are given an x-ray image of a turbine blade with a porosity. We try using the point detection algorithm to detect the porosity.

Our approach to this problem is as follows:

1. We start off by taking the following 3 x 3 kernel and convolving it with the image

0	-1	0
-1	4	-1
0	-1	0

2. We then set a threshold and check the intensity of each pixel value with the threshold. If the intensity of the pixel is less than the set threshold, we set that pixel to zero (i.e. black), if it is greater than the threshold, we set it to the maximum possible intensity (i.e. 255).

   The output when the threshold is set as 250 is as follows:

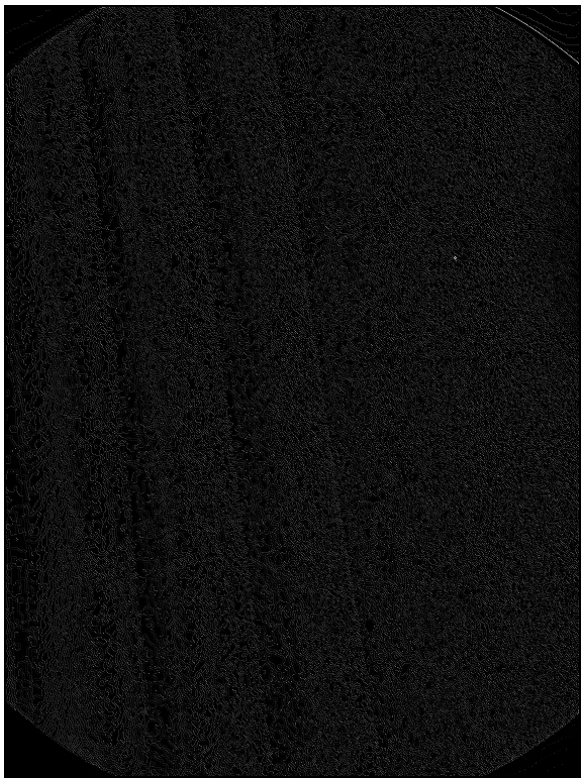

Fig 2.1 Point Detection before thresholding

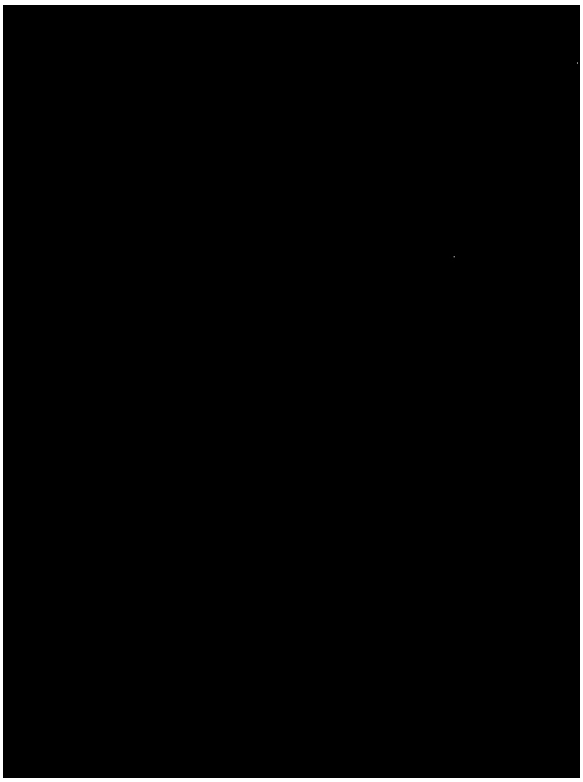

Fig 2.2 Point Detection after thresholding

We find that the location of the porosity is (249, 446) on the image.

For the part 2 of this task, we segment the object from the background using optimal thresholding. Our approach to this problem is as follows:

1. We try plotting the intensity value of pixel vs number of pixels for that intensity to find the ideal optimal threshold value.
   Our plot looks like:

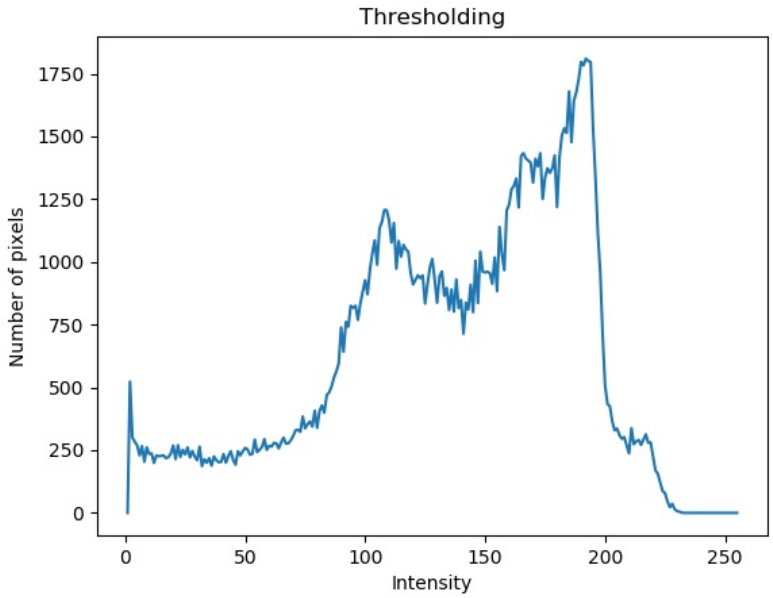

Fig 2.3 Plot of pixel intensities vs number of pixels to find optimal threshold

2. After finding the optimal threshold, we perform similar operation to make all the pixels below that threshold zero, and all above it to maximum intensity.

3. This gives us the segmented objects as follows:

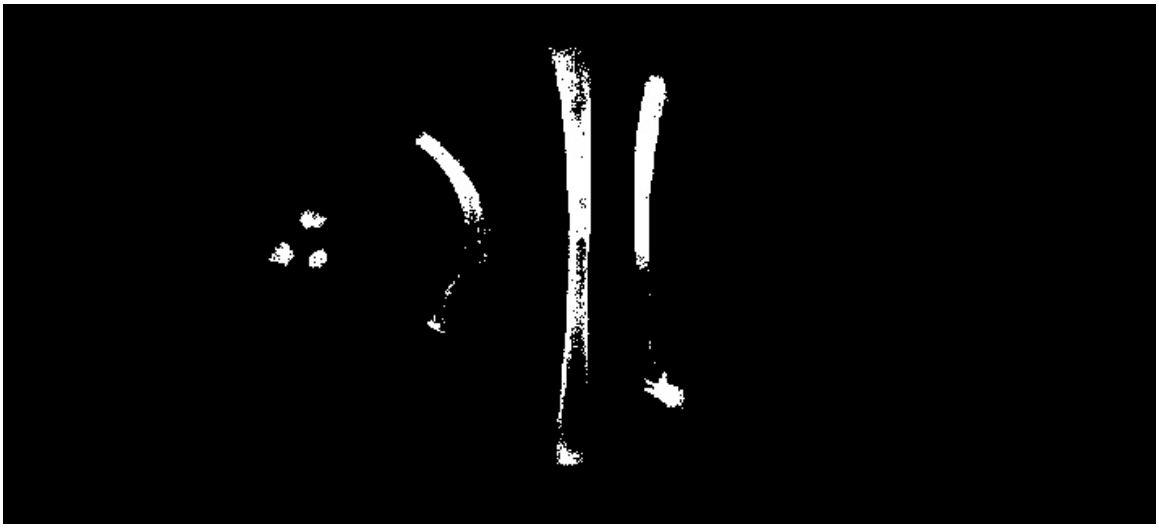

Fig 2.4 Segmented Image

4. We notice this image to have some discontinuities and hence perform morphological operations on it to improve the image and then try to find the coordinates of the object and draw a rectangle around it. We get an image as follows:

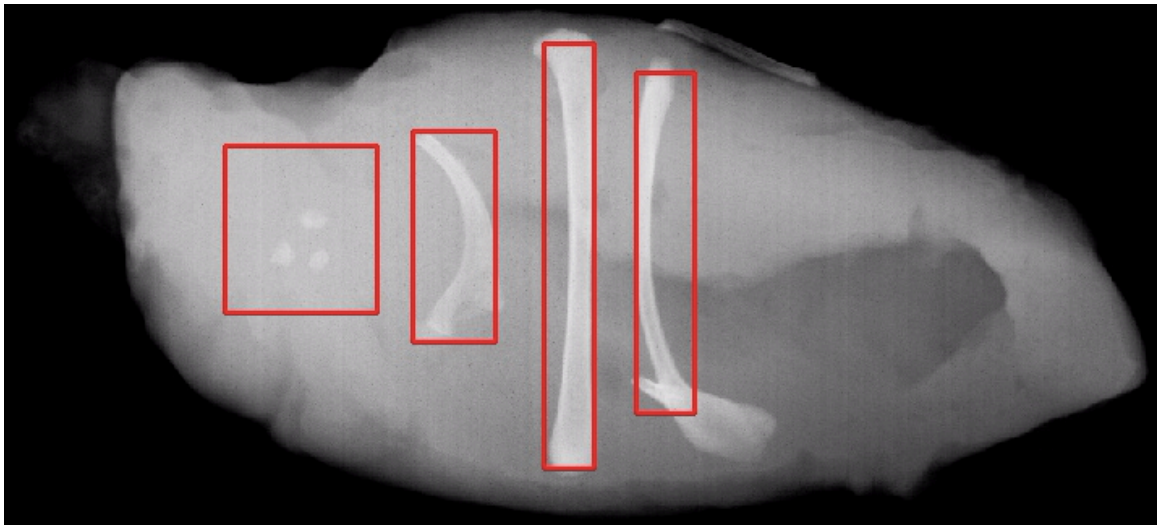

Fig 2.5 Segmented image with detected objects

We get the following coordinates of the rectangle:

(23, 332), (286, 364)
(42, 390), (252, 428)
(81, 253), (207, 302)
(87, 136), (190, 230)

Task 3: Hough Transform

The Hough transform is a feature extraction technique used in image analysis, computer vision, and digital image processing. The purpose of the technique is to find imperfect instances of objects within a certain class of shapes by a voting procedure. This voting procedure is carried out in a parameter space, from which object candidates are obtained as local maxima in a so-called accumulator space that is explicitly constructed by the algorithm for computing the Hough transform. [2]

For our task, we have the image:

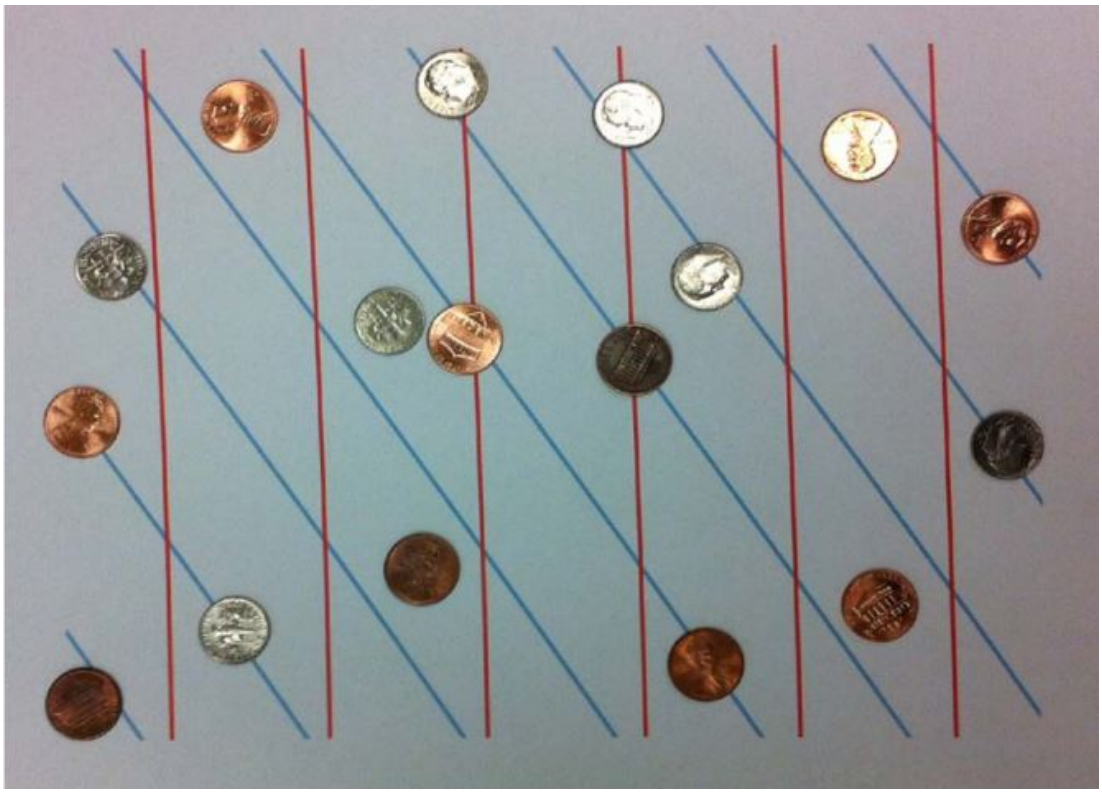

Fig 3.1 Original Image

We need to detect all the vertical lines and the diagonal lines.

Our approach to this problem is as follows:

1. We first perform Canny edge detection to find edges in the image. We get the following image as output of Canny:

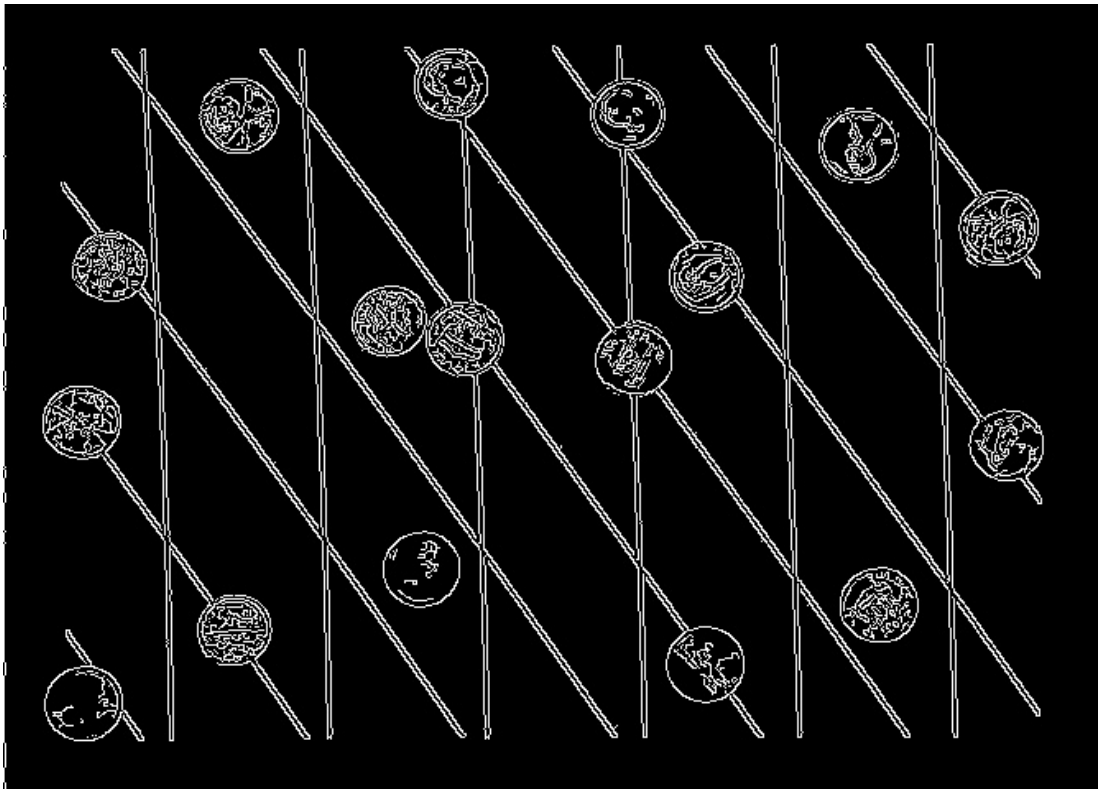

Fig 3.2 Edge image

2. We then use line detection algorithms to find the vertical and diagonal lines using the following kernels:

−1	2	−1
−1	2	−1
−1	2	−1

Fig 3.3 Vertical Line detection kernel

2	−1	−1
−1	2	−1
−1	−1	2

Fig 3.4 Diagonal Line detection kernel

3. This operation is performed to separate the diagonal and vertical lines from each other. We are able to get this by convolving the above kernel with the image and then performing morphological operations on it to remove noise. The resultant image is as follows:

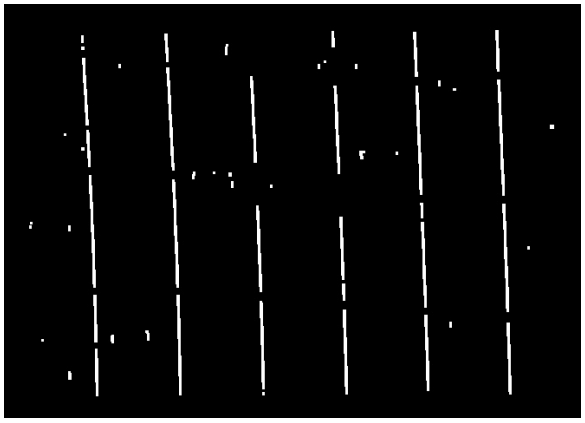

Fig 3.5 Extracted vertical lines

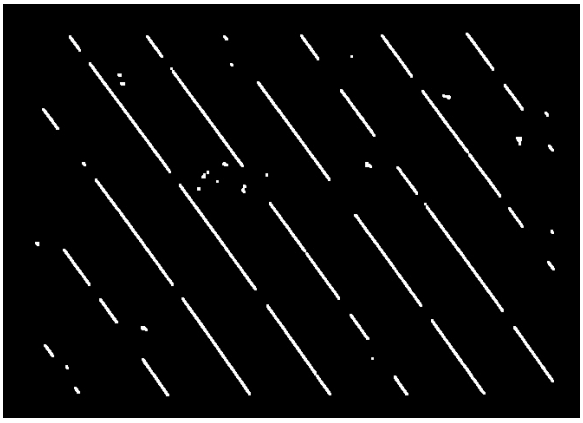

Fig 3.6 Extracted diagonal lines

4. We then build our hough space by finding each pixel in the above image and calculating its corresponding rho and theta values in the hough space using the formula:

$$x\cos\theta + y\sin\theta = r$$

Our hough space looks as follows:

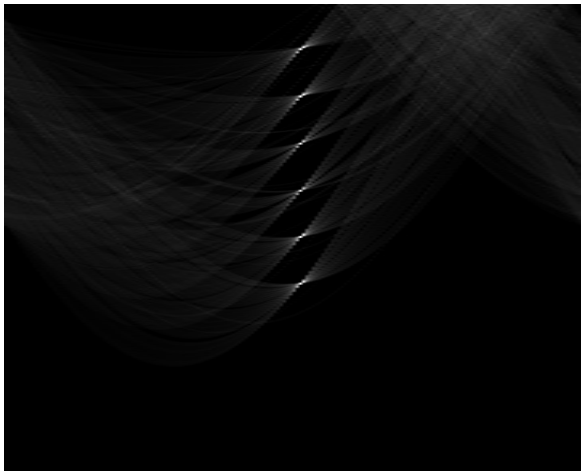

Fig 3.7 Hough space for vertical lines

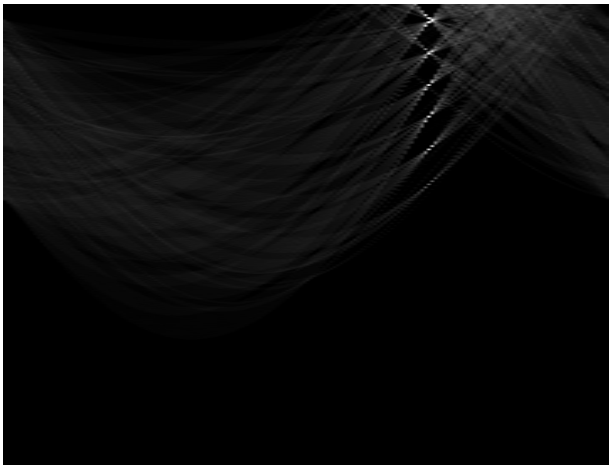

Fig 3.8 Hough space for diagonal lines

5. As we can see, we have sinusoids in our hough space. This is because each point in the image space corresponds to a sinusoid in the hough space. Hence we have each point on the lines plotted as sinusoids in the hough space.

6. We then find the point in the hough space (rho and theta values) where these sinusoids meet. And points in hough space corresponds to lines in the image space. Hence these bright spots that we see in the hough space are the lines we want in the image space.

7. When we plot these points in hough space back as lines in the image space, we observe following results:

Fig 3.9 Vertical hough lines

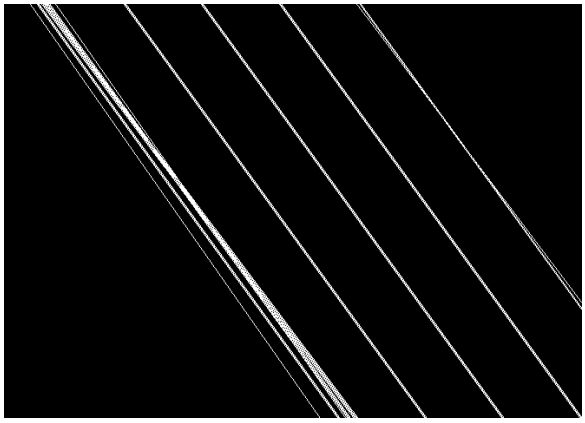

Fig 3.10 Diagonal hough lines

8. Finally, we plot these lines on the original image and get the following results:

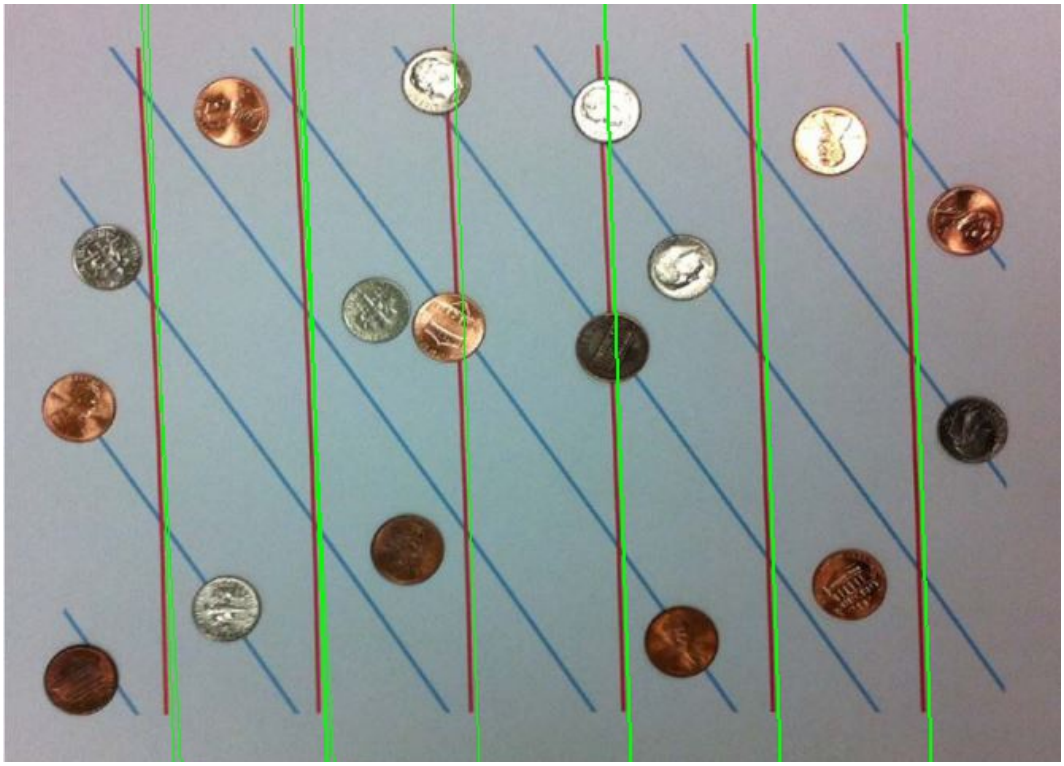

Fig 3.11 Image with vertical hough lines

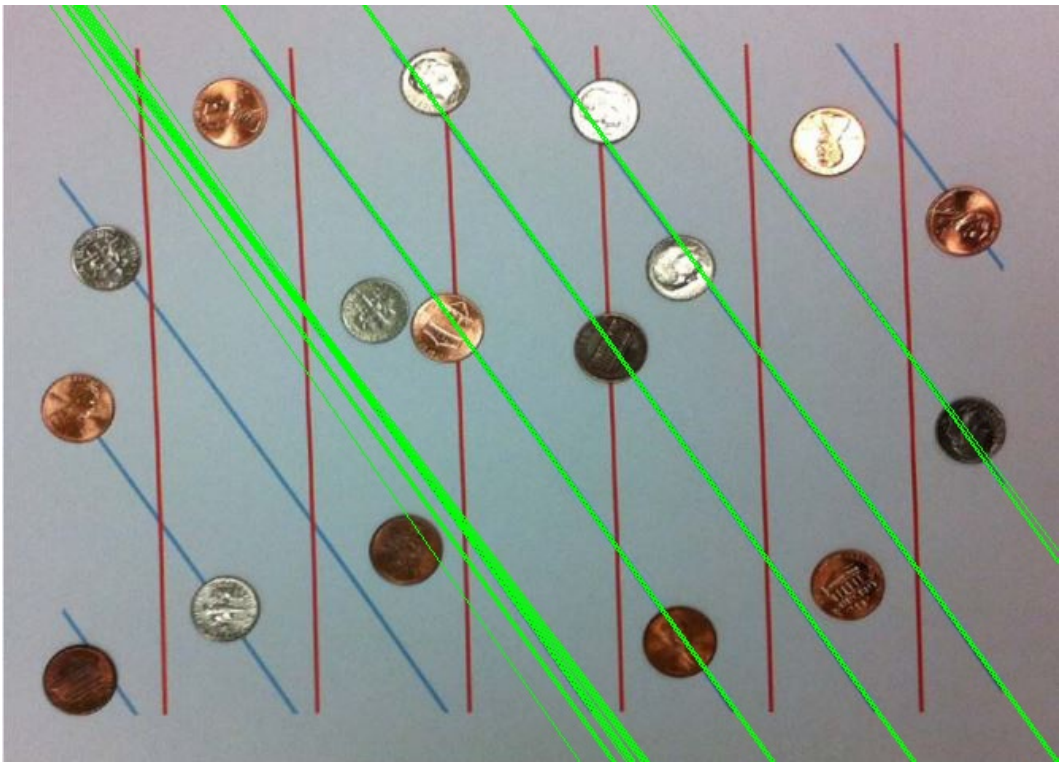

Fig 3.12 Image with diagonal hough lines

References:

1. Nick Efford
   "Digital Image Processing: A Practical Introduction using Java" – Pearson Education 2000

2. Hough Transform
   https://en.wikipedia.org/wiki/Hough_transform