# Signal Processing System Design Laboratory (EE69205)

**LAB REPORT**

by


## YASH ANAND

(Roll No.- **23EE65R22**)




**SIGNAL PROCESSING AND MACHINE LEARNING**

**Department of ELECTRICAL ENGINEERING**

**INDIAN INSTITUTE OF TECHNOLOGY KHARAGPUR**

**Kharagpur - 721302, India**


September, 2023

# || Experiment 3 ||
# || Scan conversion of multidimensional imaging signals ||

**AIM**
Implement and understand the conversion algorithms between Polar and Cartesian coordinate systems and vise versa and explore the Space complexity estimation and code profiling for executing complexity comparison.

## 3.1 Polar to Cartesian coordinate scan conversion

**FUNCTION USED:**

- **np.ravel_multi_index():** This function converts a multi-dimensional index to a linear index.
- **x_idx:** A lookup matrix of x indices for Cartesian coordinates.
- **y_idx:** A lookup matrix of y indices for Cartesian coordinates.
- **polar_image_padded:** A padded array to accommodate the polar image.

**OBSERVATION:**

→ Here when we do n dimensional convertion between polar and Cartesian coordinates, nearest-neighbor interpolation has been involved due to which here errors is being introduced, because the resolution or granularity of the grid changes during the conversion process.

→ Due to the finite representation of numbers in digital systems, here rounding errors or precision loss during calculations has been occurred. This can accumulate and result in a non-zero difference when converting back and forth.

→ The execution time for the entire process, which includes both polar-to-Cartesian and Cartesian-to-polar conversions, is approximately 0.0557 seconds.

→ The space complexity estimation for the entire process is approximately 2.2949 MB. This estimation accounts for the memory used by the images during the conversion process.

**OUTPUT:**

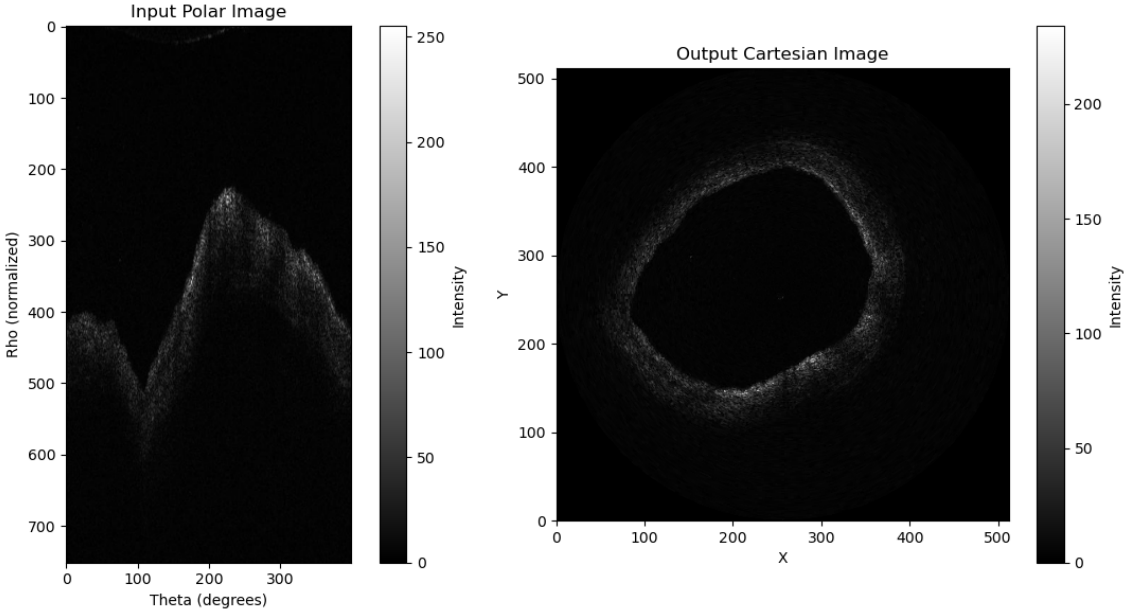


FIGURE 1: **C**onversion form input polar image to scanned cartesian image

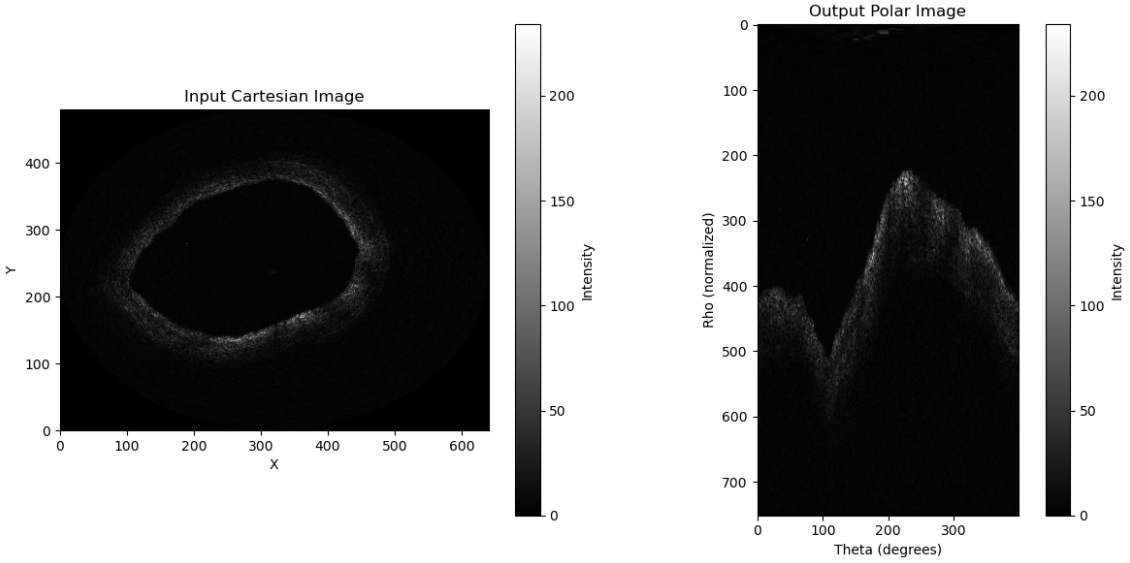


FIGURE 2: **C**onversion form scanned cartesian image to polar image

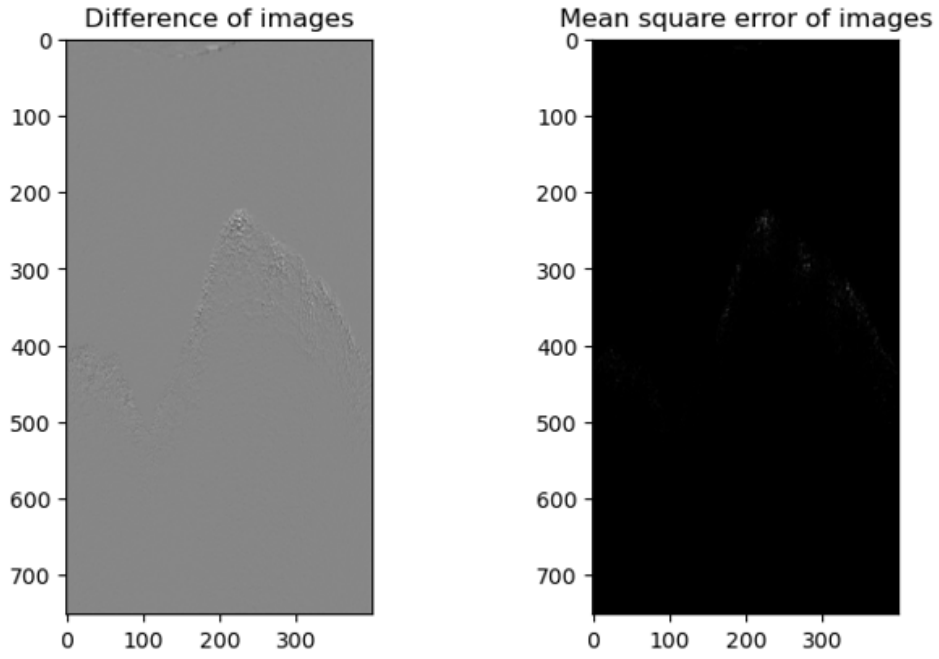


FIGURE 3: **E**rror and Mean square error difference between the original image and the transformed image.

### 3.2 Cartesian to polar scan conversion

**OBSERVATION:**

→ Here when we do n dimensional convertion between polar and Cartesian coordinates, nearest-neighbor interpolation has been involved due to which here errors is being introduced, because the resolution or granularity of the grid changes during the conversion process.

→ Due to the finite representation of numbers in digital systems, here rounding errors or precision loss during calculations has been occurred. This can accumulate and result in a non-zero difference when converting back and forth.

→ The execution time for the entire process, which includes both Cartesian-to-polar conversions and polar-to-Cartesian, is approximately 0.0581 seconds.

→ The space complexity estimation for the entire process is approximately 2.00 MB. This estimation accounts for the memory used by the images during the conversion process.

**OUTPUT:**

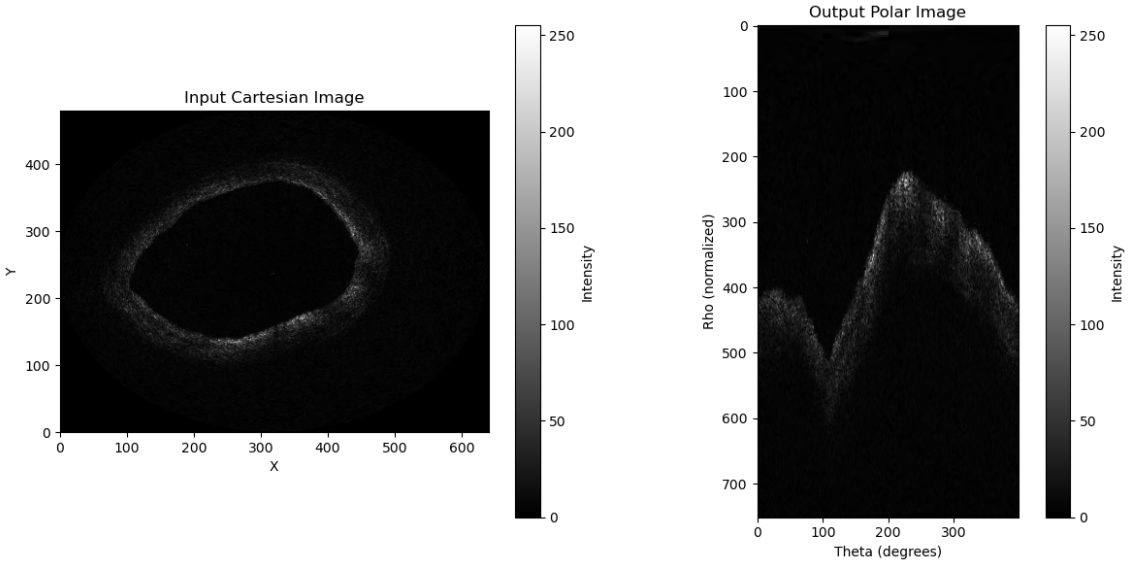


FIGURE 4: **C**onversion form input polar image to scanned cartesian image

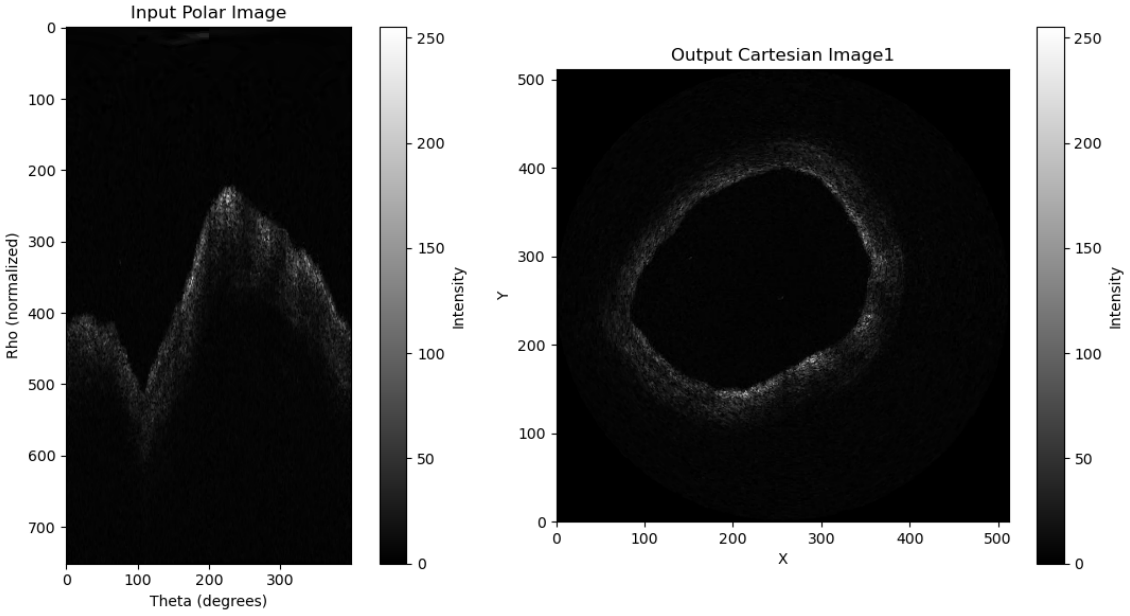


FIGURE 5: **C**onversion form scanned cartesian image to polar image

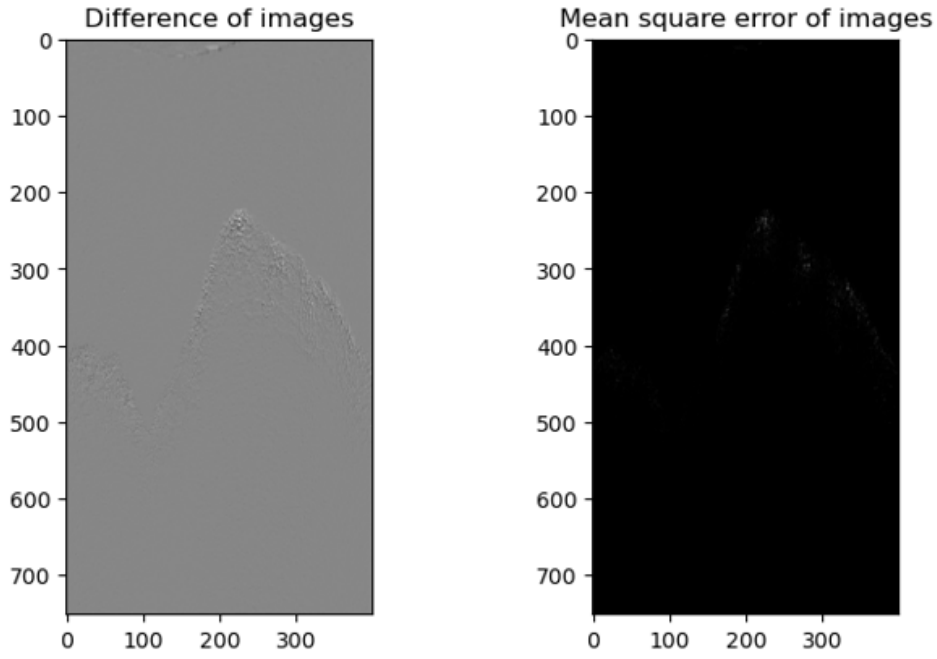


FIGURE 6: **E**rror and Mean square error difference between the original image and the transformed image.

### 3.3 Dynamic range compression in imaging signals

**OBSERVATION:**

→ The logarithmic transformation function is a non-linear function. This means that the output value is not directly proportional to the input value.

→ Logarithmic transformation compresses the image by reducing the dynamic range of the image. The dynamic range of an image is the difference between the brightest and darkest pixels in the image. A high dynamic range image has a lot of contrast between the bright and dark areas of the image, while a low dynamic range image has less contrast.

→ The logarithmic transformation has reduced the image memory size by 71%. This suggests that the logarithmic transformation has successfully compressed the image.