

Methodology:

1. **Library Imports:** The script begins by importing necessary libraries such as OpenCV for image processing, Matplotlib for visualization, NumPy for numerical operations, and other relevant modules.
2. **Initialization:** Several variables are initialized to None, including variables for storing contour data, animation objects, and image-related information.
3. **Helper Function Definitions:**
 - *f(t, t_list, x_list, y_list)*: This function performs interpolation between points (x_list, y_list) at time t.
 - *performCanny(img_gray)*: This function applies the Canny edge detection algorithm to the input grayscale image.
 - *getGrayImg()*: This function reads the input image in grayscale and displays it. Optionally, it can apply the Canny edge detection.
 - *printContours(contours)*: This function visualizes contours by plotting them.
 - *printContoursInSubplot(contours)*: This function plots contours in subplots for better visualization.
 - *findContours(img_gray)*: This function detects contours in the input grayscale image using OpenCV's findContours function.
 - *selectContour(contours)*: This function allows the user to select a contour for animation and visualization.
4. **Contour Processing:**
 - The script reads the input image, detects contours, and allows the user to select a specific contour for animation.
 - Contour points are centered around the origin and plotted.
5. **Fourier Descriptor Generation:**
 - The script generates Fourier descriptors for the selected contour using the provided generateCoefficient function.
 - Fourier descriptors capture the shape information of the contour.
6. **Animation Setup:**
 - The animation setup is initiated with the makeAnimation function, where various graphical elements are defined, such as circles, lines, and the plot area.
7. **Frame Generation:**
 - The make_frame function generates frames for the animation. It computes the position of circles based on Fourier descriptors and updates the animation accordingly.
8. **Main Function Execution:**
 - The main function orchestrates the entire process. It reads the image, processes contours, generates Fourier descriptors, sets up the animation, and saves the animation as an MP4 file.
9. **Visualization and Saving:**
 - The resulting animation is saved as an MP4 file in the specified directory.

This assignment involves image processing, contour detection, Fourier descriptor generation, animation setup, and frame generation to visualize and animate contours extracted from the input image.

Images:



Figure 4. 1: Input image (GrayScale)

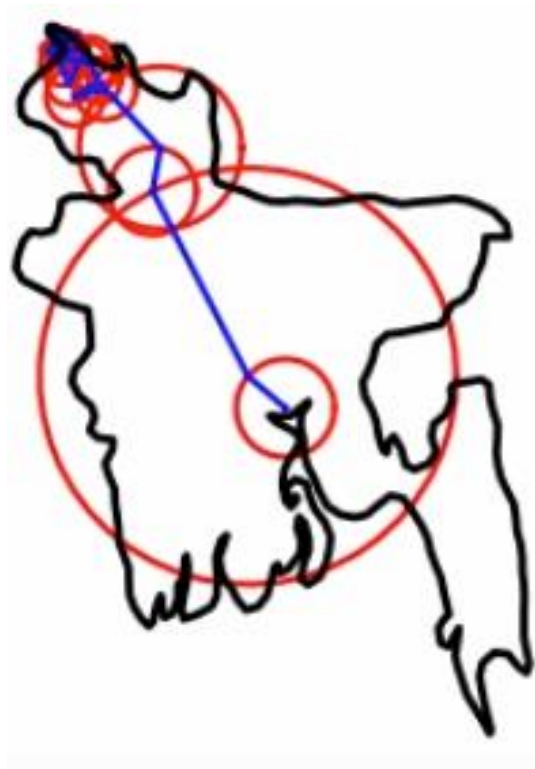


Figure 4. 2: Animated Output

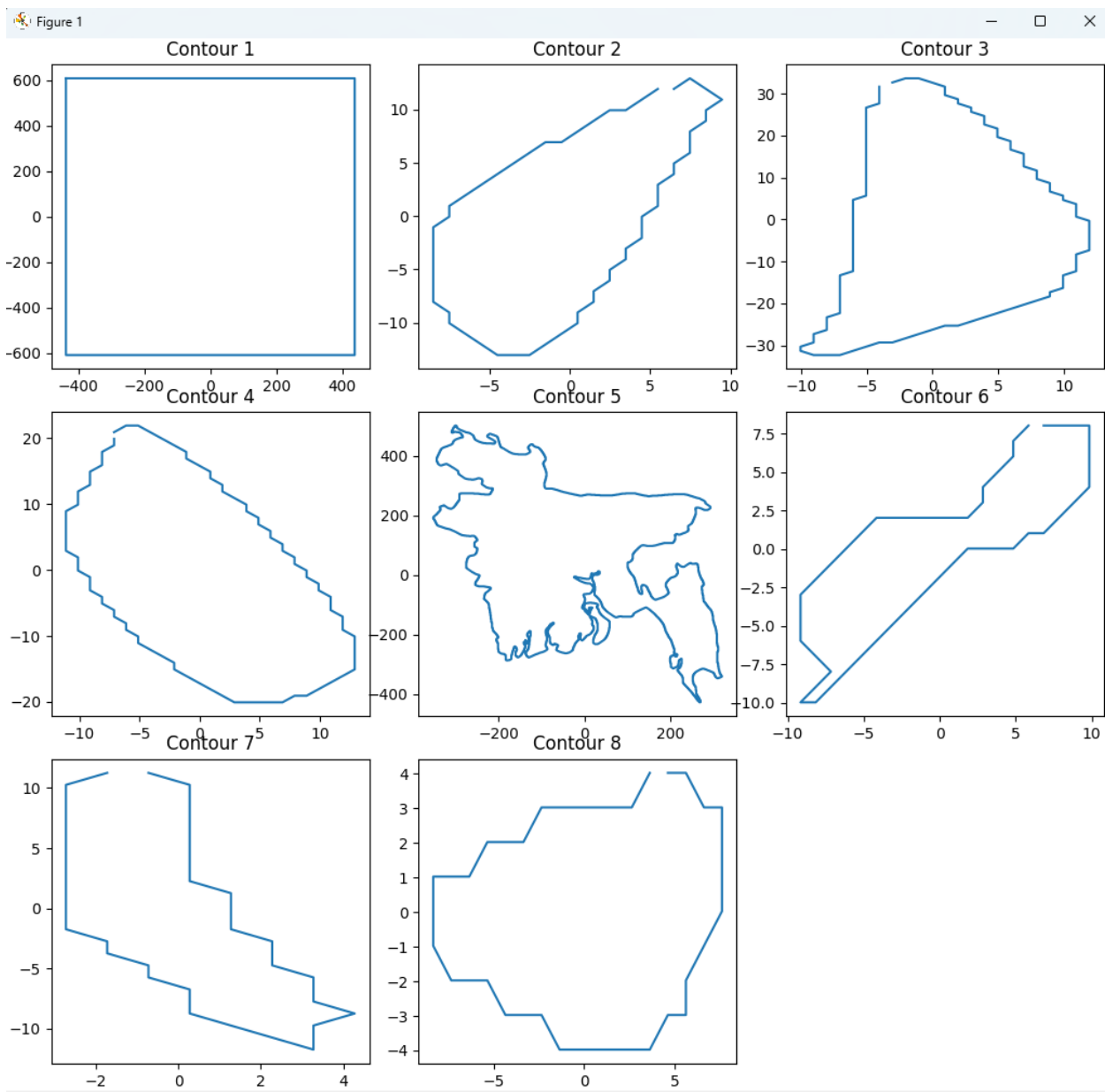


Figure 4. 3: All generated contours