**1. Interpolating and Regularizing Data**

1.1. Interpolate Points

Purpose: Interpolates a set of points to achieve a specified number of points, useful for smoothing data.

Process: Computes distances between points and interpolates new points to match the desired resolution.

1.2. Regularize Straight Lines in Polyline

Purpose: Simplifies a polyline by removing redundant points where segments are nearly collinear.

Process: Checks for collinearity and retains only necessary points.

1.3. Evaluate Accuracy of Bezier Curve

Purpose: Compares a Bezier curve to the original polyline by calculating mean squared error and maximum deviation.

Process: Interpolates original points to match Bezier curve resolution and computes error metrics.

1.4. Calculate Curvature of Bezier Curve

Purpose: Measures how curved the Bezier curve is.

Process: Computes curvature using gradients of the curve’s coordinates.

**2. Image Processing and Shape Detection**

2.1. Load and Process Image

Purpose: Processes an image to detect edges and contours.

Process: Converts the image to grayscale, applies edge detection, performs morphological operations, and finds contours.

2.2. Shape Detection Functions

Purpose: Identifies shapes in an image based on contour properties.

Process: Determines shape types (e.g., line, rectangle, circle) using contour analysis and specific shape characteristics.

2.2.1. Get Shape Name

Purpose: Classifies shapes based on the number of vertices and other attributes.

2.2.2. Detect Star Shape

Purpose: Detects star shapes based on vertex count and angles between edges.

2.2.3. Detect Ellipse

Purpose: Identifies ellipses by fitting an ellipse to the contour and checking its aspect ratio.

2.2.4. Detect Rounded Rectangle

Purpose: Detects rounded rectangles based on convexity and roundness metrics.

2.3. Draw Shapes

Purpose: Draws various shapes on an image based on detected contours.

Process: Uses OpenCV functions to draw shapes like lines, rectangles, circles, and irregular shapes.

2.4. Plotting and Display

2.4.1. Plot Paths

Purpose: Visualizes paths or lines from provided data.

Process: Plots coordinate paths on a figure.

2.4.2. Preprocess Data

Purpose: Prepares data for training and testing by organizing coordinates into feature and target arrays.

Process: Extracts and structures data for model training.

2.4.3. Build and Train Neural Network

Purpose: Creates and trains a neural network model to predict coordinates.

Process: Defines a model, compiles it, and trains it on the prepared data.

2.4.4. Evaluate Model Performance

Purpose: Assesses the trained model’s performance and visualizes the training process.

Process: Evaluates the model on test data, prints MAE, and plots training/validation loss.

2.4.5. Plot All Figures

Purpose: Provides visualizations of input data, ground truth, predictions, and regularized shapes.

Process: Creates subplots to show various aspects of the data and model results.

**3. Core Functions and Outputs for Symmetry**

3.1. CSV Data Reading

Function: read\_csv(csv\_path)

Output: Structures path data from CSV into a list of paths with segments and coordinates.

3.2. RDP Algorithm

Function: apply\_rdp(path\_XYs, epsilon=1.0)

Output: Simplifies paths by reducing points while preserving shape.

3.3. Reflection Functions

3.3.1. Reflect Points Across Vertical Line

Output: Reflected points based on the specified vertical line.

3.3.2. Reflect Points Across Horizontal Line

Output: Reflected points based on the specified horizontal line.

3.3.3. Reflect Points Across Arbitrary Line

Output: Reflected points based on the specified line and direction.

3.4. Symmetry Visualization

Function: plot\_paths\_with\_symmetry(original\_paths, reflected\_paths, colours, symmetry\_type)

Output: Visualizes original and reflected paths for different symmetry types (vertical, horizontal, diagonal).

3.5. Symmetry Evaluation

3.5.1. Function: calculate\_alignment(original\_paths, reflected\_paths)

Output: Calculates alignment score between original and reflected paths.

3.5.2. Function: find\_best\_symmetry\_axis(original\_paths)

Output: Determines the best symmetry axis angle based on alignment scores.

3.6. Testing New Data

Function: plot(path\_XYs, colours)

Output: Visualizes simplified paths for new test data.

**Bibliography:**

Sketch Recognition with Natural Correction and Editing

*Jie Wu1, ChanghuWang2, Liqing Zhang1, Yong Rui2*

*1Brain-Like Computing Lab, Shanghai Jiao Tong University, P. R. China*

*2Microsoft Research, Beijing, P. R. China*

Particle Swarm Optimization-Based Unconstrained Polygonal Fitting of 2D Shapes

*Costas Panagiotakis*

*Department of Management Science and Technology, Hellenic Mediterranean University, 72100 Agios Nikolaos, Greece*

Symmetrization of 2D Polygonal Shapes Using Mixed-Integer Programming

*Jin Huang, Jantien Stoter, Liangliang Nan*

*Urban Data Science. Delft University of Technology, 2628BL, Delft, The Netherlands*