CURVE APPROXIMATION OF NOISY DATA

Problem Statement

<u>Input:</u> noisy data of curves/lines captured by sensor
<u>Output:</u> approximation of the data by polyline with smooth curve.

Approach

To solve the problem, we have come up with a solution which satisfies the following constraints:

1. To be able to be controlled by artists we will produce the Bézier splines that fit into the given data. Control points will be given, by which artists can easily adjust the curve as their wish.
2. We guarantee that the curve will be C^1 everywhere, which means the curve we produce will have smooth shape.
3. We also guarantee that the splines will have cubic degree to fit the company's existing editing tools.
4. We will produce a slider for the artists to decide how smooth the curve they want for their tasks.
5. Since we will give the artists the liberty to decide how many points they want to control, the approximation is completely dependent on their choice. <u>Tips:</u> the more control-points you have, the better the curve will fit to the data.

Technical background

Bézier splines

As the Bézier is the heart of the approximation algorithm, we will mostly focus on how to find Bézier curve to approximate the data. The idea is that we will try to best fit the curve into the data by applying the Least-Square Approximation with the cubic Bernstein polynomials as the basis functions. The following steps will explain the algorithm in more details.

<u>Step 1:</u> Establish the Least-Square Approximations for one section of the curve

$$f = argmin\,\Big(\sum_{all\ p_i \in data} (p_i - y)^2\Big)$$

This can be derived as:

$$f = argmin\Big(\sum_{i=1}^{n}\big(p_i - (p_0.B_0 + p_1.B_1 + p_2.B_2 + p_3.B_3)\big)^2\Big) \quad (y: Bernstein\ polynomial)$$

Least Square Approximation finds the best function with the minimum distance to considered point, in other words we will find the minimum of f by solving the equation $f' = 0$.

<u>Step 2:</u> As we have already known the two control points p_0 and p_3 as they are two end points in the considered curve fragment, we then find the other two control points p_1 and p_2 that minimize f:

$$\frac{\partial f}{\partial B_1} = 0 \text{ and } \frac{\partial f}{\partial B_2} = 0$$

These yield:

$$-2.\sum_{i=1}^{n} B_1.(p_i - p_0.B_0 - p_1.B_1 - p_2.B_2 - p_3.B_3)) = 0$$
$$-2.\sum_{i=1}^{n} B_2.(p_i - p_0.B_0 - p_1.B_1 - p_2.B_2 - p_3.B_3)) = 0$$

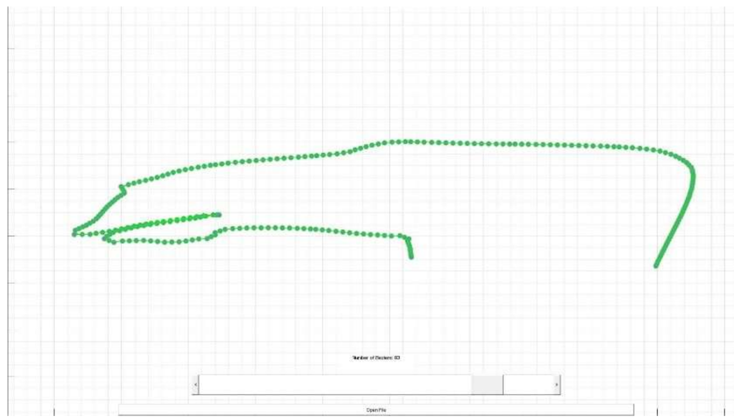
Solutions to this system of equations represented by matrix-vector form:

$$\begin{bmatrix} \sum B_1.B_1 & \sum B_1.B_2 \\ \sum B_1.B_2 & \sum B_2.B_2 \end{bmatrix} . \begin{bmatrix} p_1 \\ p_2 \end{bmatrix} = \begin{bmatrix} \sum B_1.(p_i - p_0 - p_3.B_3) \\ \sum B_2.(p_i - p_0 - p_3.B_3) \end{bmatrix}$$

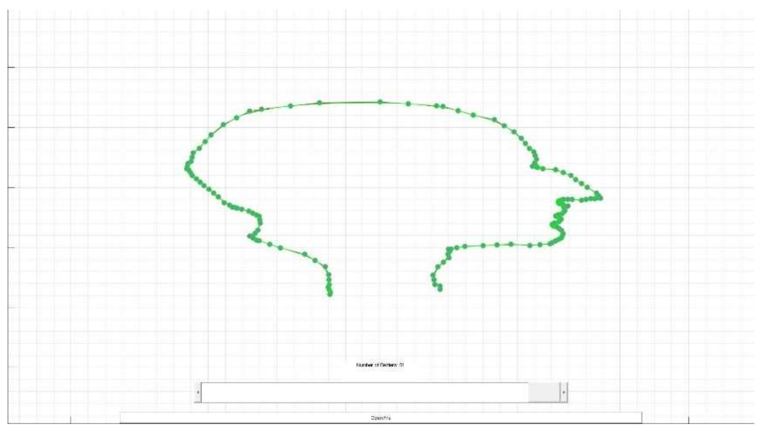
are the remaining control points p_1 and p_2.
For the C^1 continuity everywhere, Least Square Regression is applied at knot-point junction between two fragments.

Performance

<u>Test case 1:</u> CamelHeadSilhouette; no_bezier = 83



<u>Test case 2:</u> MaxPlanckSilhouette; no_bezier = 51



<u>Test case 3:</u> SineRandom; no_bezier = 28