3D Curve Development with Crossing and Twisting from 2D Drawings

AURICK DANIEL FRANCISKUS SETIADI[1], JENG WEN
JOSHUA LEAN[1], HAO-CHE KAO[1], SHIH-HSUAN HUNG[1]

[1]NATIONAL TSING HUA UNIVERSITY

Visualization & Modeling Lab

2025 TAIPEI

Motivation

2D curves with crossing points are common in ornamentation and design. These curves and their 3D counterparts are widely used in engineering and artistic applications. Twisting a 3D curve can guide textures around it as a tube or direct objects along it as a path. Conventional spline tools make it difficult to achieve specific crossings or twistings, since users must frequently change the camera view and drag control points.

Recently, VR tools such as Google's Tilt Brush allow users to draw 3D curves directly in a virtual environment. However, these tools require additional equipment and make twisting a 3D curve difficult.

OVERVIEW

Our framework transforms 2D hand-drawn sketches into 3D decorative curves. Starting from a 2D drawing, users can easily customize the behavior of crossing points and twisting. The 3D curve is then generated, allowing the user to modify its height and further decorate it. Our framework also can demonstrate elastic wire art via physics-based simulation.

METHODOLOGY

2D Curve Analysis
From 2D hand-drawn curves composed of cubic Bézier splines, we extract key feature points and identify inflection points, changes in winding direction, extreme curvature points, cusps (sharp corners) and intersection points.

3D Curve Reconstruction
We construct 3D curves with interpolating B-splines[1] that pass through the feature points captured in the input 2D curves. For each curve, users can also specify height ζ from selected point $z_i \in Z$. We also collect crossings and label them with over O, under U, or fixed-distance D. From the height and crossing constraint, we generate smooth 3D curves with a Laplacian term [SCOL*04] and solve

$$z^* = arg \min_z \sum_{i=1}^{n} \left(z_i - \sum_{j \in \mathcal{N}_i} \frac{w_{ij}}{W_i} z_j \right)^2 + \lambda \sum_{i=1}^{n} (z_i - \zeta)^2$$
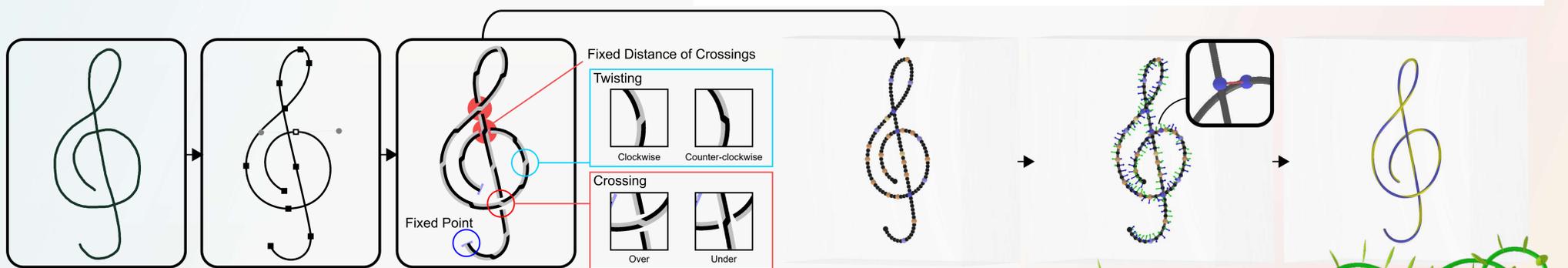
$$s.t. \quad z_i - z_j \geq d_{i,j} \quad for\ (z_i, z_j) \in O$$
$$z_i - z_j \leq -d_{i,j} \quad for\ (z_i, z_j) \in U$$
$$|z_i - z_j| = d_{i,j} \quad for\ (z_i, z_j) \in D$$
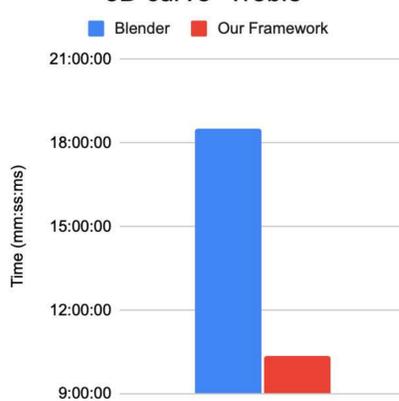$$z_i = \bar{z}_i \quad for\ z_i \in Z$$

We first compute the Laplacian term with $\mathcal{N}_i$ as neighbor feature points, w_{ij} as distance in 2D, and $W_{ij} = \sum_{j \in \mathcal{N}_i} w_{ij}$ and solve above equation with quadratic programming and then interpolate the 3D points. For the twisting, we apply linear transformation between the twist points.

Fixed Distance of Crossings

Twisting — Clockwise / Counter-clockwise
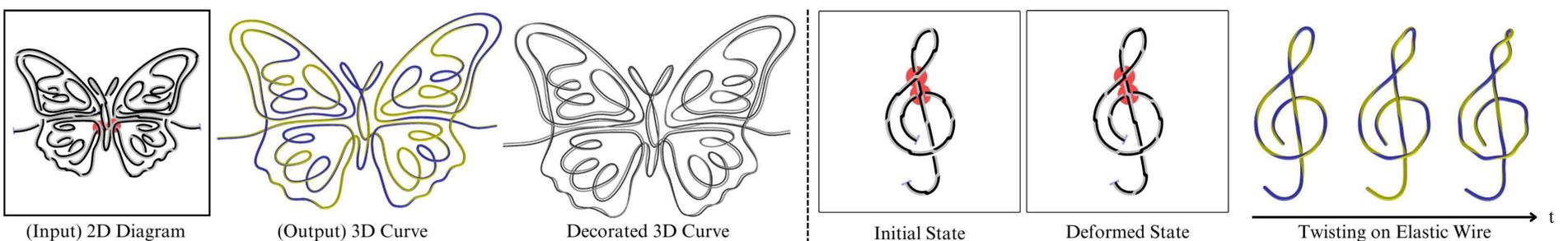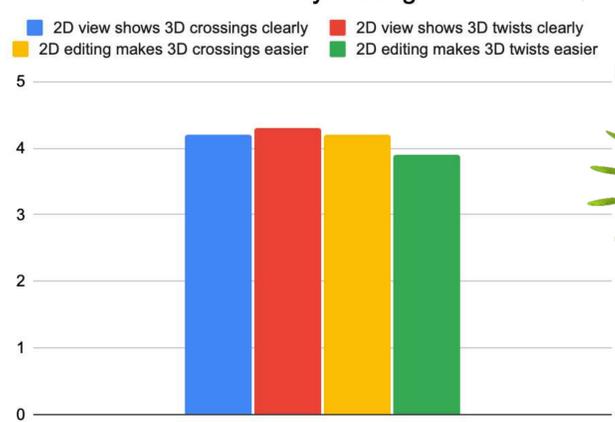
Crossing — Over / Under

Fixed Point

RESULTS

Our system enables decoration of 3D curves generated from 2D drawings and supports physics-based animations. For the computational performance, 3D curve construction averages 0.13 seconds for 20 curves with total 1,068 points. We conduct a user study to evaluate the efficiency of 3D curve development with crossings and twistings. We invited 10 participants and asked them to create 3D curves in both Blender and our system. A design is considered valid if its Fréchet distance to target curves is at most 0.5. On average, the participants spend 18 and 10 minutes on Blender and our system, respectively. We also ask the participants to rate our two-strand diagram and curve editing on a 1 to 5 scale. The clarity of crossings and twistings in our two-strand diagrams received scores of 4.2 and 4.3, respectively, while the ease of editing crossings and twistings received scores of 4.2 and 3.9.

User Study: Time taken to create 3D curve "Treble"
Blender / Our Framework

User Study: Rating
- 2D view shows 3D crossings clearly
- 2D view shows 3D twists clearly
- 2D editing makes 3D crossings easier
- 2D editing makes 3D twists easier

(Input) 2D Diagram | (Output) 3D Curve | Decorated 3D Curve | Initial State | Deformed State | Twisting on Elastic Wire

REFERENCES

[1] KNOTT G. D.: Interpolating cubic splines, vol. 18. Springer Science & Business Media, 1999. 1
[2] SORKINE O., COHEN-OR D., LIPMAN Y., ALEXA M., RÖSSL C., SEIDEL H.-P.: Laplacian surface editing. In Proceedings of the 2004 Eurographics/ACM SIGGRAPH symposium on Geometry processing (2004), pp. 175–184. 2
[3] STONE M. C., DEROSE T. D.: A geometric characterization of parametric cubic curves. ACM Transactions on Graphics (TOG) 8, 3 (1989), 147–163. 1