Sketching Curves with Immediate Feedback (sap_0360)

Sarah F. Frisk
Tufts University

1. Introduction

Computer-based drawing is important for applications ranging from sketching for illustration and design to signature capture for identity verification. While digital input devices typically provide a sequence of digitized points, applications often require input in the form of a parametric curve. Unfortunately, current methods for fitting curves to points can be complex, often require significant preprocessing of the digitized points, and can fail, especially when the path of the input points is complicated and self-intersecting. In addition, current methods typically require a full sequence of digitized points (e.g., all of the points recorded by a digital pen between pen-down and pen-up events) prior to determining any portion of the curve. This forces applications to draw an approximation of the curve until the full sequence is available (e.g., Adobe Illustrator draws the digitized points, while Microsoft’s PowerPoint draws a polyline connecting the digitized points) and may result in a noticeable delay between the pen-up event and curve generation and/or a noticeable shape change when the generated curve replaces the approximation.

We propose a method for fitting a piecewise cubic Bezier curve to a sequence of digitized points that can be performed on-the-fly rather than after the complete sequence has been acquired, thereby providing high quality parameterized curves for immediate display during drawing. The method is fast, accurate, and robust, handles complex input paths, maintains corners, and ensures at least $G^1$ continuity at non-corner points. The method uses vector distance fields to represent the intended input path of the digitized points in order to achieve its quality and performance.

Scalar distance fields represent distance as a scalar value and are used to model closed shapes with well defined insides and outsides. Vector distance fields represent distance as a vector; the 2D vector distance $(dx, dy)$ from any given sample point $(x, y)$ to the closest point $(u, v)$ on a 2D shape is the 2D vector from $(x, y)$ to $(u, v)$, i.e., $(dx, dy) = (u - x, v - y)$. Vector distance fields were introduced for evolving surfaces via level sets in [2] and for shape representation in [3]. Vector distance fields are particularly well suited for representing shapes, such as 2D lines, that do not have a well defined inside and outside because each component (e.g., $dx, dy$) of the vector distance varies smoothly, from negative to positive, from one side of the line to the other. In contrast, a scalar distance field of a line is non-differentiable along the line.

2. Efficient Curve Fitting

Most curve fitting approaches start with an initial estimating curve and iteratively adjust curve parameters to reduce a fitting error [1]. Standard approaches measure the fitting error as the average squared Euclidean distance, $d(B(t))$, from test points $B(t)$, $i = 1, N$ on the curve segment to the polyline, i.e., $E = \frac{1}{2} \sum_{i=1}^{N} d(B(t_i))^2$. The fitting error is reduced by moving the off-control curve vertices in the direction of the derivative of $E$ with respect to the positions of the off-control curve vertices. This is achieved by iteratively adjusting the control vertices $C_i$ and $C_j$ according to: $C_i = C_i^* + \alpha \cdot \frac{\nabla d(B(t_i))}{\| \nabla d(B(t_i)) \|}$ and $C_j = C_j^* + \alpha \cdot \frac{\nabla d(B(t_j))}{\| \nabla d(B(t_j)) \|}$, where $\alpha$ is usually set to 0.1. The proposed curve fitting method was compared to the widely adopted method of [5]. The proposed method proved to be faster — allowing curves to be generated during drawing rather than after a pen-up event, significantly more robust, and produced, on average, 3X fewer curves for the same accuracy and the same sequence of digitized points.

3. Results

The proposed curve fitting method was compared to the widely adopted method of [5]. The proposed method proved to be faster — allowing curves to be generated during drawing rather than after a pen-up event, significantly more robust, and produced, on average, 3X fewer curves for the same accuracy and the same sequence of digitized points.

References: