Title:
Visualization of the Curvature Monotonicity Region of 3D Bézier Curves

Authors:
Norimasa Yoshida, yoshida.norimasa@nihon-u.ac.jp, Nihon Univeristy
Takafumi Saito, txsaito@cc.tuat.ac.jp, Tokyo University of Agriculture and Technology

Keywords:
curvature monotonicity region, 3D Bézier curves, GPU

DOI: 10.14733/cadconfP.2025.1-5

Introduction:

In this work, we propose a method for visualizing the curvature monotonicity region of 3D Bézier curves. The curvature monotonicity region refers to the area in which a control point causes the curvature to change monotonically. Sapidis et al. theoretically clarified the curvature monotonicity region for quadratic Bézier curves [3], while Frey et al. extended this analysis to rational quadratic Bézier curves [1]. Yoshida et al. introduced a real-time GPU-based visualization method for the curvature monotonicity region of cubic and higher-degree Bézier curves [5]. Saito et al. further extended this approach to rational Bézier curves by introducing curvature monotonicity evaluation functions based on the Bernstein basis [2]. Prior to this work, no analysis or visualization of the curvature monotonicity region for 3D Bézier curves had been performed.

In this abstract, we extend the methods from [5, 2] to 3D polynomial and rational Bézier curves. The curvature monotonicity region exists in 3D for 3D curves. The key idea of our approach is to visualize the curvature monotonicity region on a constant-depth plane passing through the control point of interest. For 3D Bézier curves, this approach enables users to identify the region of a control point where the curvature changes monotonically.

Curvature Monotonicity Evaluation Function:

This section provides a brief review of the curvature monotonicity evaluation functions (CMEFs) introduced by Saito et al. [2]. As described in [2] for 2D curves, CMEFs help keep the fragment shader compact and efficient for GPU execution, and the same holds for 3D curves.

A 3D rational Bézier curve $\mathbf{P}(t)$ of degree $n(\geq 3)$ with $n+1$ control point vectors $\boldsymbol{p}_i = [x_i\ y_i\ z_i]^{\mathrm{T}}$ $(0 \leq i \leq n)$ and positive weights $w_i (w_i > 0)$ is

$$\mathbf{P}(t) = \frac{\sum_{i=0}^{n} B_i^n(t) w_i \boldsymbol{p}_i}{\sum_{i=0}^{n} B_i^n(t) w_i} = \frac{\boldsymbol{Q}(t)}{W(t)}, \tag{2.1}$$

where $B_i^n(t)$ is the Bernstein polynomial of degree n. We assume that the curve is regular and that the curvature is nonzero. If all the weights w_i are 1, the curve is a polynomial Bézier curve. For a 3D curve $\mathbf{P}(t)$, the derivative of the curvature with respect to the arc length shown in [4] is

$$\frac{d\kappa}{ds} = \frac{\left((\dot{\mathbf{P}} \wedge \dddot{\mathbf{P}}) \cdot (\dot{\mathbf{P}} \wedge \ddot{\mathbf{P}})\right)(\dot{\mathbf{P}} \cdot \dot{\mathbf{P}}) - 3\left((\dot{\mathbf{P}} \wedge \ddot{\mathbf{P}}) \cdot (\dot{\mathbf{P}} \wedge \ddot{\mathbf{P}})\right)(\dot{\mathbf{P}} \cdot \ddot{\mathbf{P}})}{(\dot{\mathbf{P}} \cdot \dot{\mathbf{P}})^3 |\dot{\mathbf{P}} \wedge \ddot{\mathbf{P}}|}, \tag{2.2}$$

where $\cdot$ and $\wedge$ are dot and cross products, and $\dot{\mathbf{P}}$, $\ddot{\mathbf{P}}$ and $\dddot{\mathbf{P}}$ are the first, second, and third derivatives of $\mathbf{P}(t)$ with respect to t. We write the numerator of Eq. (2.2) as

$$L_n(t) = \left((\dot{\mathbf{P}} \wedge \dddot{\mathbf{P}}) \cdot (\dot{\mathbf{P}} \wedge \ddot{\mathbf{P}})\right)(\dot{\mathbf{P}} \cdot \dot{\mathbf{P}}) - 3\left((\dot{\mathbf{P}} \wedge \ddot{\mathbf{P}}) \cdot (\dot{\mathbf{P}} \wedge \ddot{\mathbf{P}})\right)(\dot{\mathbf{P}} \cdot \ddot{\mathbf{P}}). \tag{2.3}$$

With the assumption that the curve is regular and the curvature is nonzero, the denominator of Eq. (2.2) is always positive. Therefore, the curvature monotonicity can be evaluated using $L_n(t)$. For rational curves, $L_n(t)$ is a rational function whose denominator is always positive. Thus, the curvature monotonicity is determined by the numerator of $L_n(t)$, which we denote as $l_n(t)$ (See Eq. (2.7)). We refer to $L_n(t)$ or $l_n(t)$ as the curvature monotonicity evaluation function.

Saito et al. derived equations for $L_n(t)$ and $l_n(t)$ in Bernstein basis [2]. We use these Bernstein basis equations to visualize the curvature monotonicity regions of 3D curves. To represent $L_n(t)$ and $l_n(t)$ in the Bernstein basis, we utilize internal division points and weights of de Casteljau's algorithm. For $0 \leq k \leq m \leq n$, k-th internal division point $\boldsymbol{Q}_{m,k}(t)$ and weight $W_{m,k}(t)$ at $(n-m)$-th step are given by:

$$\boldsymbol{Q}_{m,k}(t) = \sum_{i=0}^{n-m} B_i^{n-m} w_{k+i} \boldsymbol{p}_{k+i}, \qquad W_{m,k}(t) = \sum_{i=0}^{n-m} B_i^{n-m} w_{k+i}. \tag{2.4}$$

If both m and k are single-digit integers, the comma between them is omitted, as in Q_{01}. The advantage of the $\boldsymbol{Q}_{m,k}$ expression is that expressions involving derivatives up to the m-th order can be derived using $\boldsymbol{Q}_{i,k}$ $(1 \leq i \leq m)$ for curves of degree n.

For a 3D polynomial curve $\mathbf{P}(t)$ of degree $n (n \geq 3)$,

$$L_n(t) = (\boldsymbol{V}_4 \cdot \boldsymbol{V}_3)(\boldsymbol{V}_1 \cdot \boldsymbol{V}_1) - 3(\boldsymbol{V}_3 \cdot \boldsymbol{V}_3)(\boldsymbol{V}_1 \cdot \boldsymbol{V}_2), \tag{2.5}$$

where

$$\begin{aligned}
\boldsymbol{V}_1 &= n\ (\boldsymbol{Q}_{11} - \boldsymbol{Q}_{10}), \qquad \boldsymbol{V}_2 = n(n-1)\ (\boldsymbol{Q}_{22} - 2\boldsymbol{Q}_{21} + \boldsymbol{Q}_{20}),\\
\boldsymbol{V}_3 &= n^2(n-1)\ (\boldsymbol{Q}_{20} \wedge \boldsymbol{Q}_{21} + \boldsymbol{Q}_{22} \wedge \boldsymbol{Q}_{20} + \boldsymbol{Q}_{21} \wedge \boldsymbol{Q}_{22}),\\
\boldsymbol{V}_4 &= n^2(n-1)(n-2)\left((1-t)(\boldsymbol{Q}_{31} - \boldsymbol{Q}_{30}) \wedge (2\boldsymbol{Q}_{31} - 3\boldsymbol{Q}_{32} + \boldsymbol{Q}_{33})\right.\\
&\qquad \left. +\quad t\quad (\boldsymbol{Q}_{30} - 3\boldsymbol{Q}_{31} + 2\boldsymbol{Q}_{32}) \wedge (\boldsymbol{Q}_{33} - \boldsymbol{Q}_{32})\right),
\end{aligned}$$

and the curvature monotonicity can be evaluated with the degree $6n-11$ function $L_n(t)$.

In the degree n rational space Bézier curve $\mathbf{P}$ $(n \geq 3)$,

$$L_n(t) = \frac{l_n(t)}{(W(t))^{11}}, \tag{2.6}$$

where

$$\begin{aligned}
l_n(t) &= n(((\boldsymbol{V}_5 \wedge \boldsymbol{V}_6) \cdot \boldsymbol{V}_7)(\boldsymbol{V}_5 \cdot \boldsymbol{V}_5) + 3(\boldsymbol{V}_7 \cdot \boldsymbol{V}_7)(\boldsymbol{V}_8 \cdot \boldsymbol{V}_5)),\\
\boldsymbol{V}_5 &= n\ (W_{10}\boldsymbol{Q}_{11} - W_{11}\boldsymbol{Q}_{10}),\\
\boldsymbol{V}_6 &= n(n-1)(n-2)(W_{30}\boldsymbol{Q}_{33} - 3W_{31}\boldsymbol{Q}_{32} + 3W_{32}\boldsymbol{Q}_{31} - W_{33}\boldsymbol{Q}_{30}),\\
\boldsymbol{V}_7 &= n(n-1)(W_{22}(\boldsymbol{Q}_{20} \wedge \boldsymbol{Q}_{21}) + W_{21}(\boldsymbol{Q}_{22} \wedge \boldsymbol{Q}_{20}) + W_{20}(\boldsymbol{Q}_{21} \wedge \boldsymbol{Q}_{22})),\\
\boldsymbol{V}_8 &= n^2(n-1)\ ((1-t)(2W_{11}(W_{20}\boldsymbol{Q}_{21} - W_{21}\boldsymbol{Q}_{20}) - W_{10}(W_{20}\boldsymbol{Q}_{22} - W_{22}\boldsymbol{Q}_{20}))\\
&\qquad +\quad t\quad (W_{11}(W_{20}\boldsymbol{Q}_{22} - W_{22}\boldsymbol{Q}_{20}) - 2W_{10}(W_{21}\boldsymbol{Q}_{22} - W_{22}\boldsymbol{Q}_{21})),
\end{aligned} \tag{2.7}$$

and the curvature monotonicity can be evaluated with the degree $11n-18$ function $l_n(t)$.

In 3D polynomial or rational curves, $L_n(t)$ or $l_n(t)$ can be expressed as a polynomial of degree n_c in Bernstein form:

$$\lambda(t) = \sum_{i=0}^{n_c} B_i^{n_c}(t)\xi_i. \tag{2.8}$$

For 3D polynomial curves, $n_c = 6n - 11$ and $\lambda(t)$ corresponds to Eq. (2.5). For 3D rational curves, $n_c = 11n - 18$ and $\lambda(t)$ corresponds to Eq. (2.7).

The curvature monotonicity can be evaluated by checking if $\lambda(t)$ changes its sign within $t \in [0, 1]$. We refer to the condition described below as the exact curvature monotonicity condition, or simply *the exact condition*.

$$\lambda(t) \leq 0 \ \text{ or } \ \lambda(t) \geq 0 \ \text{ for } t \in [0, 1] \tag{2.9}$$

If $\lambda(t) \leq 0$ for $t \in [0, 1]$, the curvature is monotonically decreasing. Conversely, if $\lambda(t) \geq 0$ $t \in [0, 1]$, the curvature is monotonically increasing. We refer to the following condition as the *sufficient condition*.

$$\xi_i \leq 0 \ (0 \leq i \leq n_c) \ \text{ or } \ \xi_i \geq 0 \ (0 \leq i \leq n_c) \tag{2.10}$$

From the convex hull property, it is evident that if the sufficient condition is satisfied, the exact condition is also satisfied. Note that the exact condition may be satisfied even if the ξ_is have different signs. We visualize the curvature monotonicity regions of 3D curves using the exact and sufficient conditions.

Visualization of Curvature Monotonicity Regions of 3D Bézie rcurves :

In 3D curves, the curvature monotonicity region of a control point exists in 3D space. We propose a method to visualize this region by displaying its intersection with a 2D plane. The region can be interactively explored by adjusting the viewing parameters.

Let $\boldsymbol{p}_j$ be the control point for which we wish to visualize the curvature monotonicity region. Let T_p represent the matrix that transforms a point from world coordinates to screen coordinates. The matrix T_p is the product of the viewing, projection, and viewport transformations. By multiplying $\boldsymbol{p}_j$, expressed in homogeneous coordinates, by T_p, we can obtain the depth d of $\boldsymbol{p}_j$. We visualize the curvature monotonicity region of $\boldsymbol{p}_j$ as its intersection with a plane at depth d.

Similarly to [5, 2], we visualize the curvature monotonicity region by checking the curvature monotonicity for every pixel in a screen window using a GPU. Algorithm 1 presents a method for checking curvature monotonicity, implemented in a fragment shader using OpenGL.

Algorithm 1. Curvature monotonicity region of $\boldsymbol{p}_j$

(1) Using the screen coordinates along with the depth d, we perform the inverse transformation T_p^{-1} to obtain the 3D coordinates $\boldsymbol{p}'_j$.

(2) Replace $\boldsymbol{p}_j$ with $\boldsymbol{p}'_j$, and check the curvature monotonicity using either Eq. (2.5) or (2.7), following a similar algorithm to Algorithm 1 in [5].

(3) If the curvature is monotonically varying, the pixel is colored with the user-specified color.

To visualize the curvature monotonicity regions of all control points simultaneously, Algorithm 1 is repeated for each control point within the fragment shader.

Fig. 1 shows the curvature monotonicity regions of a cubic Bézier curve with $\boldsymbol{p}_0 = [0\ 0\ 0]^{\mathrm{T}}$, $\boldsymbol{p}_1 = [0.2\ 0\ 0]^{\mathrm{T}}$, $\boldsymbol{p}_2 = [0.6\ 0.2\ 0.1]^{\mathrm{T}}$, $\boldsymbol{p}_3 = [0.8\ 1\ 0.2]$. In Fig. 1(a), the curve is polynomial, as all weights are 1. Note that the depth of each region, corresponding to each control point, is different. In the upper left of the figure, the curvature plot is shown. Fig. 1(b) and (c) show the same curve from different viewpoints. Since we visualize the curvature monotonicity region as the intersection with a constant-depth plane, the

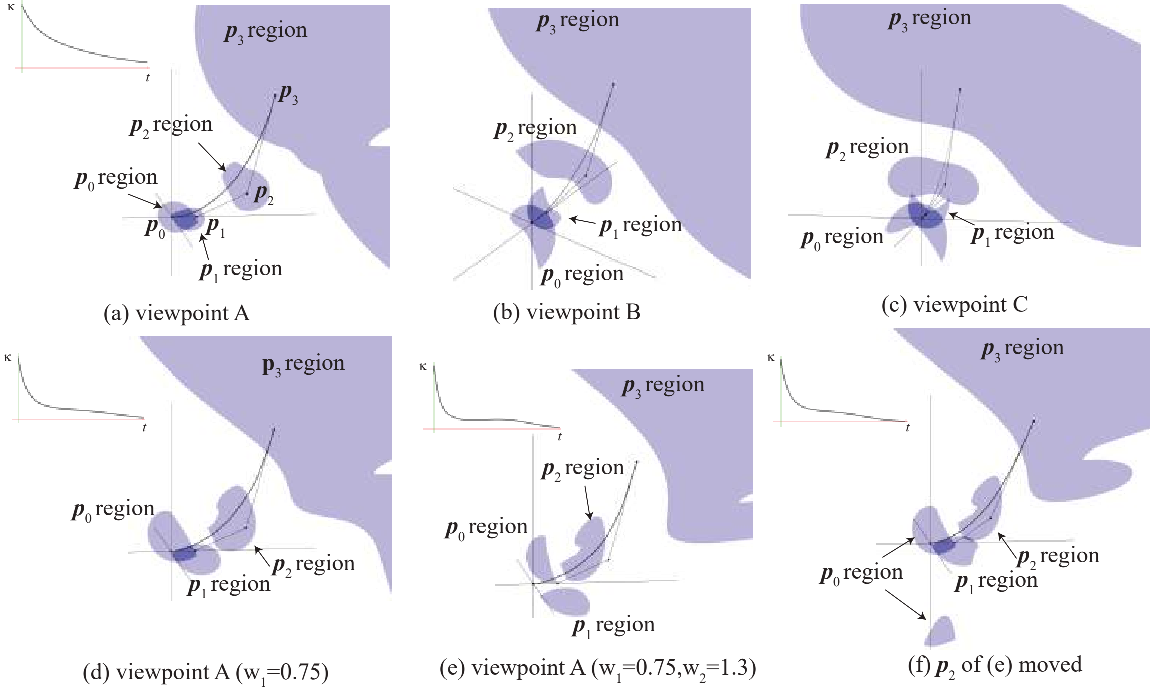

Fig. 1: Curvature monotonicity region of cubic Bézier curves

shape of the region varies depending on the viewpoint. In Fig. 1(d), with the same viewpoint A, w_1 is set to 0.75 and the curvature remains monotonically decreasing. In Fig. 1(e), w_2 is set to 1.3, where it can be observed that the curvature is no longer monotonically decreasing. In Fig. 1(f), $\boldsymbol{p}_2$ is moved within the sufficient region so that the curvature becomes monotonically decreasing.

Fig. 2 shows the sufficient region of each control point with the same control points as in Fig. 1 (a) but from a different viewpoint. In each figure, the regions where $\xi_i > 0$ are colored. Therefore, the white region represents the area where all ξ_i are negative, meaning that the curvature will be monotonically decreasing if the control point is placed within this region. Based on the properties of the Bernstein polynomial, if ξ_0 or ξ_{n_c} (which is ξ_7 in this case) defines the boundary of the white region, they also define the boundary of the exact region.

Conclusions:

In this abstract, we proposed a method for visualizing the curvature monotonicity regions of 3D Bézier curves on a constant-depth plane that passes through the control point of interest. Using a GPU, real-time visualization of exact and sufficient regions is possible for lower-degree curves. Our approach enables users to generate curves with monotonically varying curvature more easily since the region of a control point where the curvature varies monotonically can be visualized. For future work, we plan to focus on visualizing the curvature monotonicity region of a 3D Bézier curve as a 3D surface.

Acknowledgement:

This work was supported by JSPS KAKENHI Grant Number 24K07280.

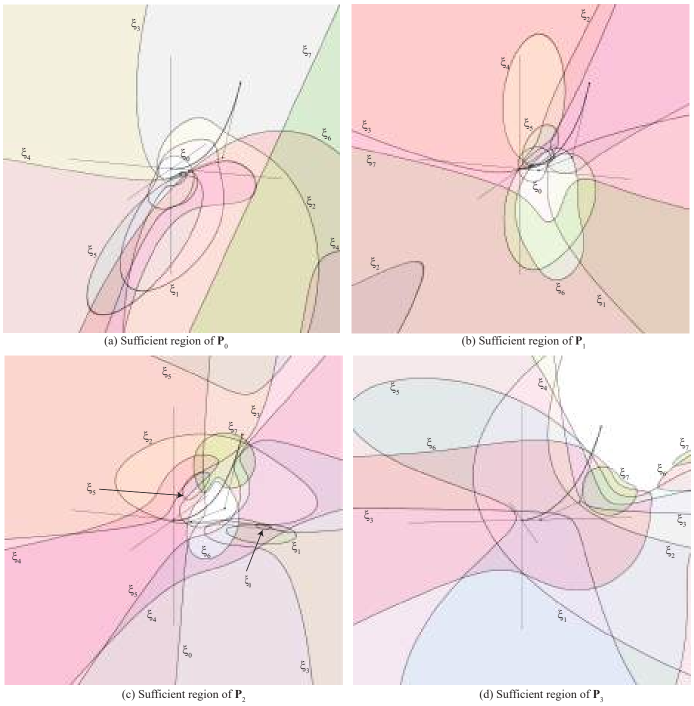

Fig. 2: Sufficient regions as the intersection of implicit regions

Norimasa Yoshida, https://orcid.org/000-0000-1234-5678
Takafumi Saito, https://orcid.org/0000-0001-5831-596X

References:

[1] Frey, W. H.; Field, D.A.: Designing Bézier conic segments with monotone curvature. Computer Aided Geometric Design, 17(6), 457-483, 2000. http://dx.doi.org/10.1016/S0167-8396(00)00011-X

[2] Saito, T.; Yoshida, N.: Curvature monotonicity evaluation functions on rational Bézier curves, Computers & Graphics, 114,2023, 219-229. https://doi.org/10.1016/j.cag.2023.05.019

[3] Sapidis, N. S.; Frey, W. H.: Controlling the curvature of a quadratic Bézier curve, Computer Aided Geometric Design, 9, 1992, 85-91. https://doi.org/10.1016/0167-8396(92)90008-D

[4] Yoshida, N., Fukuda, R., Saito, T.: Logarithmic Curvature and Torsion Graphs, In Mathematical Methods for Curves and Surfaces, MMCS 2008, Lecture Notes in Computer Science, vol 5862, pp.434-443, 2010. https://doi.org/10.1007/978-3-642-11620-9_28

[5] Yoshida, N.; Sakurai, S.; Yasuda, H.; Inoue, T.; Saito, T.: Visualization of the Curvature Monotonicity Regions of Polynomial Curves and its Application to Curve Design, Computer-Aided Design and Applications, 21(1), 2024, 75-87. https://doi.org/10.14733/cadaps.2024.75-87

[6] Yoshida, N.; Saito, T.: Curvature Monotonicity Regions of 2D Polynomial and Rational Bézier Curves as the Intersection of Implicit Regions, Computer-Aided Design and Applications, 22(1), 2025, 68-80. https://doi.org/10.14733/cadaps.2025.68-80