

Übung zu Computergraphik II

– Übungsblatt 6 –

Lehrstuhl für Computergraphik
und Multimediasysteme

Peter Marchel, Julian Bader, Hendrik Hochstetter

Aufgabe 1 [1 Punkt] Auswerten von TP-Bézierflächen

Gegeben sei die bi-quadratische Bézier-Fläche mit den Kontrollpunkten

$$\begin{aligned} \mathbf{C}_{00} &= \begin{pmatrix} 0 \\ 2 \\ 2 \end{pmatrix}, & \mathbf{C}_{10} &= \begin{pmatrix} 4 \\ 2 \\ 6 \end{pmatrix}, & \mathbf{C}_{20} &= \begin{pmatrix} 12 \\ 2 \\ 2 \end{pmatrix}, & \mathbf{C}_{01} &= \begin{pmatrix} 0 \\ 6 \\ 6 \end{pmatrix}, & \mathbf{C}_{11} &= \begin{pmatrix} 4 \\ 6 \\ 10 \end{pmatrix} \\ \mathbf{C}_{21} &= \begin{pmatrix} 12 \\ 6 \\ 6 \end{pmatrix}, & \mathbf{C}_{02} &= \begin{pmatrix} 0 \\ 14 \\ 2 \end{pmatrix}, & \mathbf{C}_{12} &= \begin{pmatrix} 4 \\ 14 \\ 6 \end{pmatrix}, & \mathbf{C}_{22} &= \begin{pmatrix} 12 \\ 14 \\ 2 \end{pmatrix} \end{aligned}$$

1. Werten Sie die Fläche mittels des zweistufigen de Casteljau Algorithmus für den Parameter (0.5,0.5) zweimal aus:
 - Zuerst in u - und dann in v -Richtung
 - Anschließend erst in v - und dann in u -Richtung
2. Bestimmen Sie die Flächennormale $\hat{n}$ für die Parameter (0,0) und (1,1).
3. Bestimmen Sie die Flächennormale $\hat{n}$ für den Parameter (0.5,0.5).

Aufgabe 2 [1 Punkt] Programm Frenet-Frame

Schreiben Sie ein Programm zur Visualisierung des Frenet-Frames einer 3D-Bézier-Kurve.

Das zur Verfügung stehende Programm hat folgende Funktionalität:

1. Daten einlesen per Argument (z.B. data_01): Grad n sowie die entsprechenden Bézier-Punkte, Anzahl der Samplepunkte für die Anzeige.
2. Auswerten der 3D-Bézierkurve für gegebenen Parameter.
3. Kurve und Frenet-Frame visualisieren.

Das bereitgestellte Programm bestehend aus folgenden Klassen:

P3D, V3D: Eine 3D-Punkt- und Vektorklasse inkl. verschiedener Operatoren zur Manipulation von Punkten und Vektoren.

`ParamCurve3D`: Die bereits definierte Klasse für 3D-Kurven und weiteren Methoden für die Ermittlung der ersten und zweiten Ableitung.

`BezierCurve3D`: Die konkrete Instanz einer 3D-Parameterkurve

`FrenetFrame`: Klasse zur Berechnung des Frenet-Frames für eine gegebene `ParamCurve3D`.

`DisplayOGL`: Klasse zur Anzeige der Kurve und des Frames für einen festen Parameterwert.

Die Klassenhierarchie ist vollständig, folgende Methoden und Operatoren wurden allerdings noch nicht implementiert:

1. Kreuzprodukt `V3D::operator^(const V3D& vec) const`.

2. Berechnung der ersten und zweiten Ableitung in

`BezierCurve3D::eval(double u, bool preserveDerivatives)`.

Hinweis:

Die Ableitung berechnen sich anhand der berechneten Punkte beim de Casteljau-Algorithmus.

$$\mathbf{C}'(u) = n(\mathbf{C}_1^{n-1}(u) - \mathbf{C}_0^{n-1}(u)), \quad \mathbf{C}''(u) = n(n-1)((\mathbf{C}_2^{n-2}(u) - \mathbf{C}_1^{n-2}(u)) - (\mathbf{C}_1^{n-2}(u) - \mathbf{C}_0^{n-2}(u)))$$

3. Berechnung der Vektoren des Frenet-Frames

`bool FrenetFrame::computeFrame(double u, P3D& position, V3D& tangent, V3D& normal, V3D& binormal) .`

Abgabe: 25.11.2013, zu Beginn der Vorlesung oder bis 10:00 Uhr im Postkasten des Lehrstuhls (gegenüber Raum H-A 7107)