

Interdisziplinäres Projekt

Glatte Kurven aus diskreten Punkten

Carlos Camino

`cfcamino@mytum.de`

16. Oktober 2013

Ziele dieser Arbeit

1. Reduzierte Datenmenge
2. Effiziente Berechnung
3. Ästhetische Darstellung

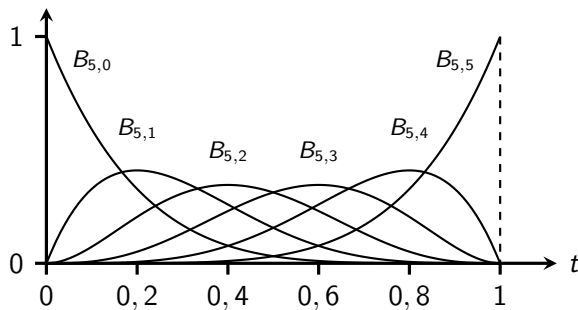
Bernsteinpolynome

Definition

Sei $n \in \mathbb{N}_0$ und $i \in \{0, \dots, n\}$. Das i -te **Bernsteinpolygon** n -ten Grades hat die Gestalt

$$B_{n,i}(t) := \binom{n}{i} \cdot t^i \cdot (1-t)^{n-i}.$$

Beispiel



Bézierkurven

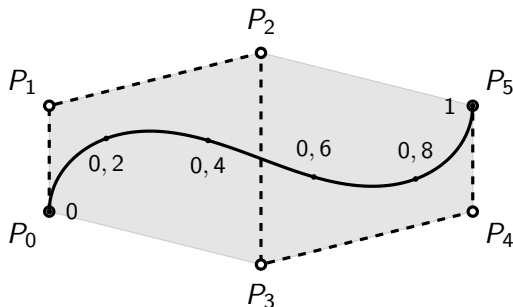
Definition

Seien $P_0, \dots, P_n \in \mathbb{R}^d$ beliebige Punkte. Eine Kurve $C = \{X(t) \mid t \in [0, 1]\}$ mit

$$X(t) = \sum_{i=0}^n B_{n,i}(t) \cdot P_i$$

heißt **Bézierkurve** n -ten Grades.

Beispiel



B-Spline-Basisfunktionen

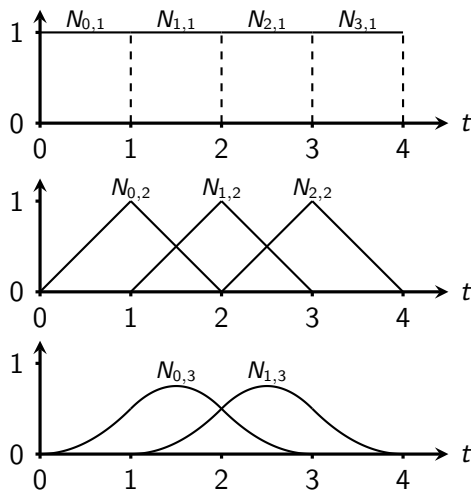
Definition

Für einen gegebenen Vektor $T = (t_0, t_1, \dots, t_{m-1})$ der Länge m sind die **B-Spline-Basisfunktionen** rekursiv definiert:

$$N_{i,k}(t) = \begin{cases} I_{[t_i, t_{i+1})}(t) & \text{falls } k = 1 \\ \frac{t - t_i}{t_{i+k-1} - t_i} \cdot N_{i,k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} \cdot N_{i+1,k-1}(t) & \text{sonst} \end{cases}$$

Beispiel

Basisfunktionen für $T = (0, 1, 2, 3, 4)$ und $k = 1, 2, 3$:



B-Splines

Definition

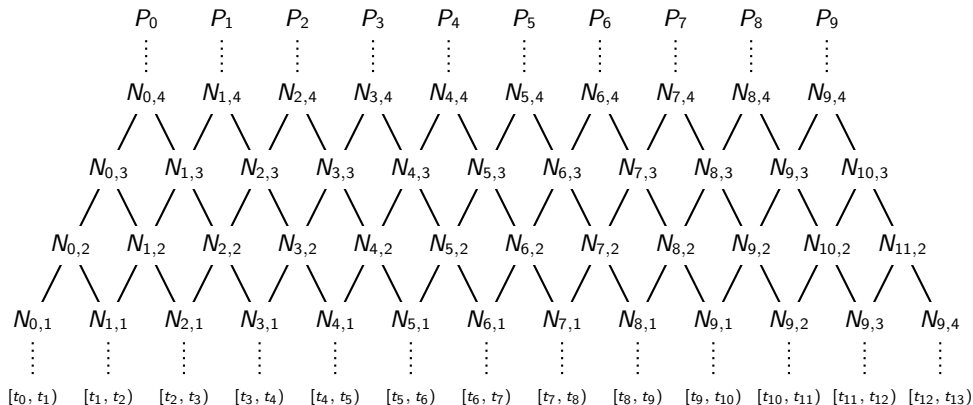
Seien $P_0, P_1, \dots, P_{n-1} \in \mathbb{R}^d$ beliebige Punkte und $T = (t_0, t_1, \dots, t_{m-1})$ ein Knotenvektor. Eine Kurve $C = \{X(t) \mid t \in [t_{k-1}, t_{m-k})\}$ mit

$$X(t) = \sum_{i=0}^{n-1} N_{i,k}(t) \cdot P_i$$

heißt **B-Spline** der **Ordnung** k .

Beispiel

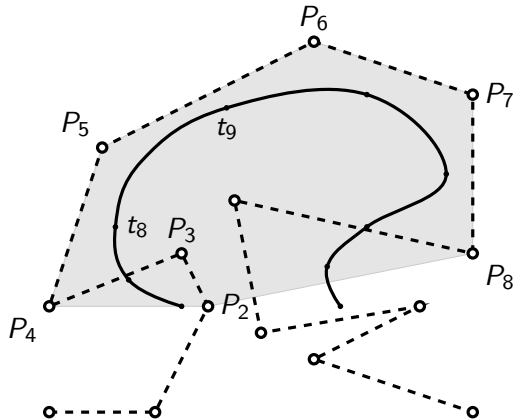
Berechnungsschema für $k = 4$, $n = 10$ und $m = 14$:



Es gilt immer:

$$m = n + k .$$

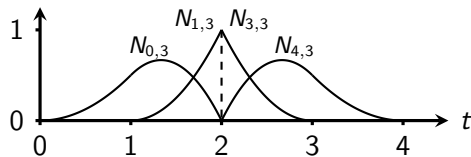
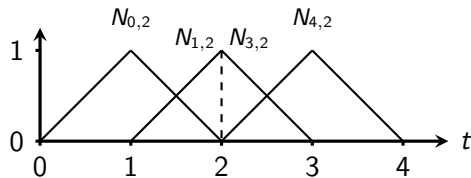
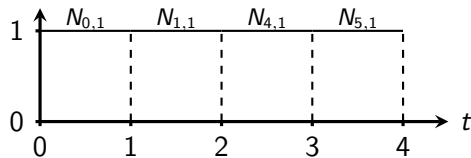
Beispiel



Mehrfachknoten

Beispiel

Basisfunktionen für $T = (0, 1, 2, 2, 2, 3, 4)$ und $k = 1, 2, 3$:



Eingespannte B-Splines

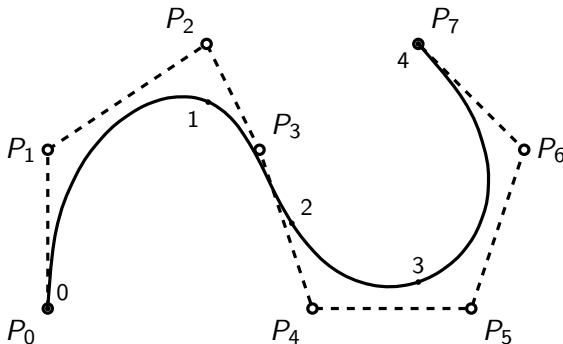
Eingespannte B-Splines haben einen Knotenvektor der Form

$$T = (\underbrace{t_0, \dots, t_{k-1}}_{\text{gleiche Knoten}}, \underbrace{t_k, \dots, t_{m-k-1}}_{\text{innere Knoten}}, \underbrace{t_{m-k}, \dots, t_{m-1}}_{\text{gleiche Knoten}}).$$

Für sie gilt $N_{0,k}(t_{k-1}) = 1$ und $N_{n-1,k}(t_{m-k}) = 1$.

Beispiel

B-Spline mit $k = 5$ und $T = (0, 0, 0, 0, 0, 1, 2, 3, 4, 4, 4, 4, 4)$:



Einfügen von Knoten

Ziel: Füge t_i in $T = (t_0, \dots, t_{i-1}, t_{i+1}, \dots, t_m)$ hinzu.

Methode:

1. Berechnung von Hilfsvariablen $l_0, \dots, l_k$ mithilfe der Formel:

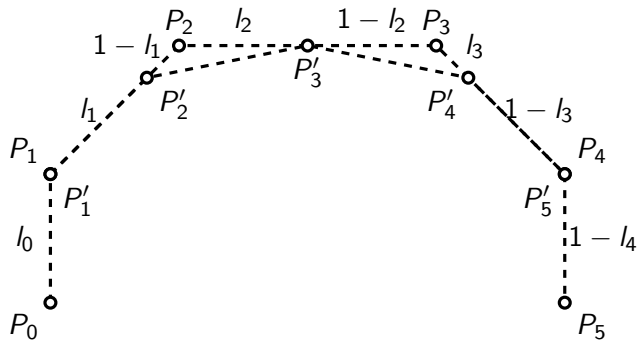
$$l_j = \frac{t_i - t_{i-k+j}}{t_{i+j} - t_{i-k+j}} \quad (j = 0, \dots, k) .$$

2. Berechnung von Hilfspunkten $P'_{i-k}, \dots, P'_i$. Es gilt:

$$P'_{j+i-k} = (1 - l_j) \cdot P_{j+i-k-1} + l_j \cdot P_{j+i-k} \quad (j = 0, \dots, k) .$$

3. Ersetzen der Punkte $P_{i-k}, \dots, P_{i-1}$ durch die Hilfspunkte $P'_{i-k}, \dots, P'_i$.

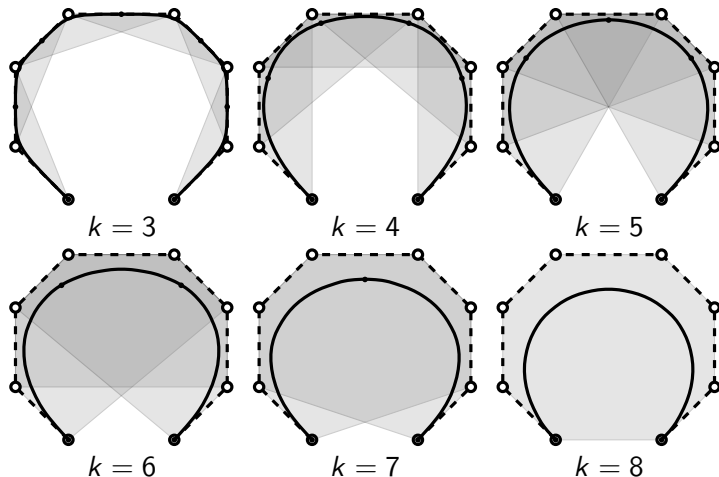
Beispiel



Konsequenzen der Lokalitätseigenschaft

Die konvexen Hüllen von B-Splines sind kleiner als die von Bézierkurven. Die konvexe Hülle einer Bézierkurve entspricht der desjenigen B-Splines mit Ordnung $k = n$.

Beispiel



Implementierung des Fensters

Idee: Benutze Fenster mit fester Größe $> 2k$.

Algorithmus:

1. Speichere Knoten im Fenster mit aktueller Anzahl w an Punkten.
2. Wenn das Fenster voll ist oder keine Punkte mehr dazukommen, lösche in den $w - 2k$ inneren Spalten des Fensters so viele Punkte wie es geht.
3. Füge übriggebliebenen Punkte in den approximierten B-Spline hinzu und bewege das Fenster.

Wie wählt man t_i ?

1. Chord-Length-Parametrisierung:

$$t_i = t_{i-1} + \frac{1}{k-1} \sum_{j=i-k+1}^{i-1} |\overline{P_{j-1}P_j}|$$

2. Event-Time-Parametrisierung:

$$t_i = t_{i-1} + \frac{1}{k-1} \sum_{j=i-k+1}^{i-1} e_j$$

Verwendete Variablen:

Variable	Bedeutung
windowCapacity	Obere Grenze für die die Anzahl der aktiven Knoten im Fenster, d.h. für $w - 2k$
order	Ordnung k des B-Splines
useEventTime	Event-Time (true) oder Chord-Length (false)
winX[i]	x -Koordinate des Punktes P_i im Fenster
winY[i]	y -Koordinate des Punktes P_i im Fenster
winE[i]	Event time Differenz e_i des Punktes P_i im Fenster
winT[i]	Knoten t_i im Fenster
winN	Aktuelle Anzahl w der Punkte im Fenster
bspX[i]	x -Koordinate des Punktes P_i in der approximierten Kurve
bspY[i]	y -Koordinate des Punktes P_i in der approximierten Kurve
bspT[i]	Knoten t_i in der approximierten Kurve
bspN	Aktuelle Anzahl der Punkte in der approximierten Kurve

Löschen von Knoten

Erinnerung: Einfügen von t_i in T

1. Zuerst bestimme $P'_{i-k}, \dots, P'_i$ mit $P'_{i-k+j} = (1 - l_j) \cdot P_{i-k+j-1} + l_j \cdot P_{i-k+j}$ und $l_j = \frac{t_i - t_{i-k+j}}{t_{i+j} - t_{i-k+j}}$.
2. Dann ersetze $P_{i-k}, \dots, P_{i-1}$ durch $P'_{i-k}, \dots, P'_i$.

Als Matrix:

$$P' = A \cdot P$$

mit $P = (P_{i-k}, \dots, P_{i-1})^T$, $P' = (P'_{i-k}, \dots, P'_i)^T$ und

$$A = \begin{pmatrix} l_0 & 0 & 0 & \dots & 0 \\ 1 - l_1 & l_1 & 0 & \dots & 0 \\ 0 & 1 - l_2 & l_2 & \dots & 0 \\ 0 & 0 & 1 - l_3 & \dots & 0 \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & \dots & 1 - l_k \end{pmatrix}.$$

Neues Ziel: Lösche t_r von T .

Ansatz: Finde $S = (S_{r-k}, \dots, S_{r-1})$, so dass

$$\|P - A \cdot S\|_p$$

minimal ist und ersetze P durch S .

Spezielle Lösungen:

$$Q_j = \frac{1}{l_{k-r+j}} \cdot P_j + \left(1 - \frac{1}{l_{k-r+j}}\right) \cdot Q_{i-1} \quad (r-k \leq j \leq r-1)$$

und

$$R_j = \frac{1}{l_{k-r+j+1}} \cdot P_{j+1} + \left(1 - \frac{1}{l_{k-r+j+1}}\right) \cdot R_{i+1} \quad (r-k \leq j \leq r-1)$$

Allgemeine Lösung:

$$S_j = (1 - \mu_{k-r+j}) \cdot Q_j + \mu_{k-r+j} \cdot R_j \quad (r-k \leq j \leq r-1).$$

Satz

$S = (S_{r-k}, \dots, S_{r-1})^T$ ist genau dann eine Lösung von $\min_S \|P - A \cdot S\|_\infty$, wenn $\mu_0, \dots, \mu_{k-1}$ die Gleichung

$$\mu_j = \frac{1}{\gamma_\infty} \cdot \sum_{i=0}^j \begin{bmatrix} k-1 \\ i+1 \end{bmatrix} \quad (0 \leq j \leq k-1)$$

mit $\gamma_\infty = \sum_{i=0}^k \begin{bmatrix} k-1 \\ i+1 \end{bmatrix}$ erfüllen. Insbesondere gilt dann $w_r = \|P - A \cdot S\|_\infty = \frac{\|D\|_2}{\hat{\gamma}_\infty}$.

Bemerkung:

Die Werte von $\begin{bmatrix} k-1 \\ 1 \end{bmatrix}, \begin{bmatrix} k-1 \\ 2 \end{bmatrix}, \dots, \begin{bmatrix} k-1 \\ k+1 \end{bmatrix}$ und D lassen sich aus den Werten von $t_{r-k}, \dots, t_{r+k}$ berechnen.

Satz

$S = (S_{r-k}, \dots, S_{r-1})^T$ ist genau dann eine Lösung von $\min_S \|P - A \cdot S\|_2$, wenn $\mu_0, \dots, \mu_{k-1}$ die Gleichung

$$\mu_j = \frac{1}{\gamma_2} \cdot \sum_{i=0}^j \left[\begin{matrix} k-1 \\ i+1 \end{matrix} \right]^2 \quad (0 \leq j \leq k-1)$$

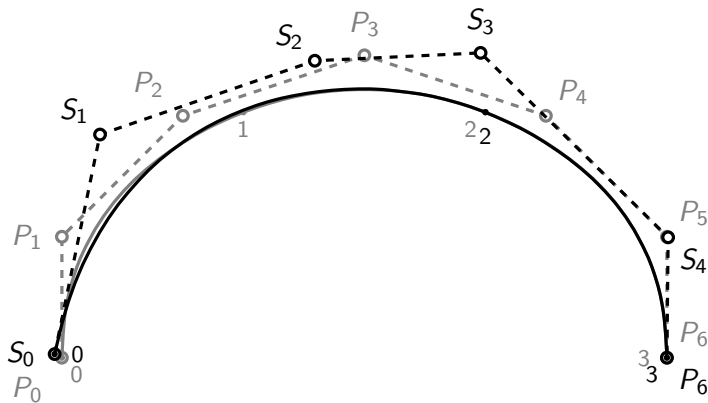
mit $\gamma_2 = \sum_{i=0}^k \left[\begin{matrix} k-1 \\ i+1 \end{matrix} \right]^2$ erfüllen. Insbesondere gilt dann $w_r = \|P - A \cdot S\|_2 = \frac{\|D\|_2}{\sqrt{\hat{\gamma}_2}}$.

Bemerkung:

Die Werte von $\left[\begin{matrix} k-1 \\ 1 \end{matrix} \right], \left[\begin{matrix} k-1 \\ 2 \end{matrix} \right], \dots, \left[\begin{matrix} k-1 \\ k+1 \end{matrix} \right]$ und D lassen sich aus den Werten von $t_{r-k}, \dots, t_{r+k}$ berechnen.

Beispiel

Eingepannter B-Spline der Ordnung $k = 5$ mit $n = 7$ Kontrollpunkten und Knotenvektor $T = (0, 0, 0, 0, 0, 1, 2, 3, 3, 3, 3, 3)$ vor (grau) und nach (schwarz) dem Löschen des Knotens 1:

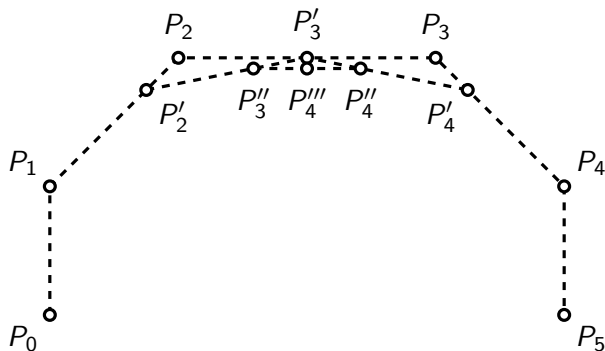


Verwendete Variablen:

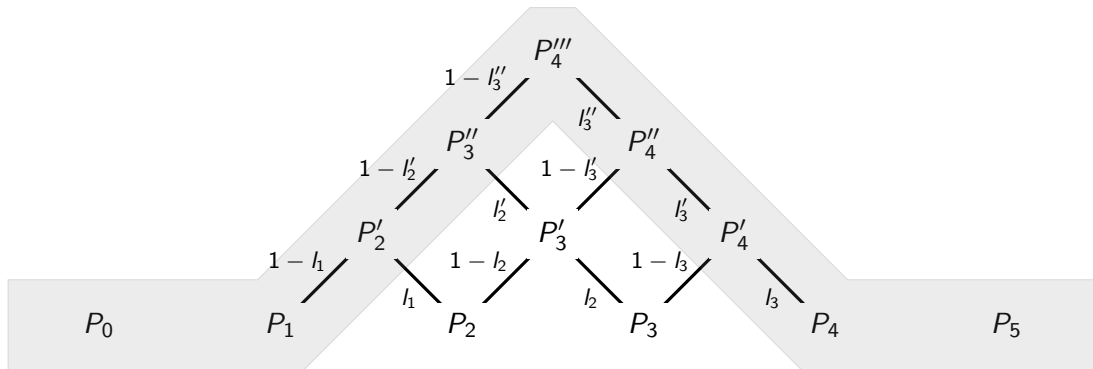
Variable	Bedeutung
maxAlteration	Obere Grenze für die Veränderung der Kurve (je größer, desto mehr Knoten werden gelöscht)
useNormL2	Wahl der diskreten Approximation: L_∞ oder L_∞
auxL[i][j]	Wert von l_j für den Knoten t_i
auxB[i][j]	Wert von $\begin{bmatrix} k-1 \\ j+1 \end{bmatrix}$ für den Knoten t_i
auxD[i]	Wert von $\ D\ _2$ für den Knoten t_i
auxG[i]	Wert von γ_2 bzw. von γ_∞ für den Knoten t_i
auxW[i]	Wert von w_i für den Knoten t_i
auxM[i][j]	Wert von μ_j für den Knoten t_i
auxQX[j]	x-Koordinate von Q_{j+r-k} für den zu löschenden Knoten
auxQY[j]	y-Koordinate von Q_{j+r-k} für den zu löschenden Knoten
auxRX[j]	x-Koordinate von R_{j+r-k} für den zu löschenden Knoten
auxRY[j]	y-Koordinate von R_{j+r-k} für den zu löschenden Knoten
auxSX[j]	x-Koordinate von S_{j+r-k} für den zu löschenden Knoten
auxSY[j]	y-Koordinate von S_{j+r-k} für den zu löschenden Knoten

Darstellung als Bézierkurven

Der Algorithmus von de Boor erlaubt es einen Knoten mehrmals in T einzufügen und berechnet die Kontrollpunkte nach dem Einfügen.



Die Berechnung der Punkte erfolgt nach folgendem Schema:



Man nennt dies Anordnung **de Boer-Netz**

Hat ein Knoten Multiplizität $k - 1$, so kann der Knotenvektor

$$T = (t_0, \dots, t_{i-1}, \underbrace{t_i, \dots, t_{i+k-2}}_{\text{gleiche Knoten}}, t_{i+k-1}, \dots, t_{m-1})$$

in

$$T_1 = (t_0, \dots, \underbrace{t_i, \dots, t_{i+k-2}}_{\text{gleiche Knoten}}) \quad \text{und} \quad T_2 = (\underbrace{t_i, \dots, t_{i+k-2}}_{\text{gleiche Knoten}}, \dots, t_{m-1}) .$$

zerlegt werden.

Schließlich kann man die Kontrollpunkte eines B-Splines mit Knotenvektor

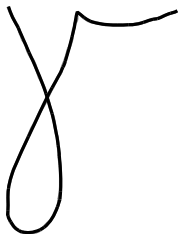
$$T = (\underbrace{t_0, \dots, t_{k-1}}_{\text{gleiche Knoten}}, \underbrace{t_k, \dots, t_{2k-1}}_{\text{gleiche Knoten}})$$

direkt als Kontrollpunkte für eine Bézierkurve übernommen werden.

Verwendete Variablen:

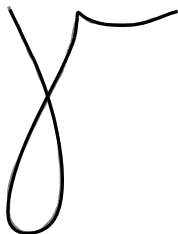
Variable	Bedeutung
netL[a][b]	Wert von $I_a^{(b)}$ im de Boor Netz
netX[a][b]	x-Koordinate von $P_{a+i-k+1}^{(b)}$ im de Boor Netz
netY[a][b]	y-Koordinate von $P_{a+i-k+1}^{(b)}$ im de Boor Netz
bezX[i][j]	x-Koordinate des i -ten Kontrollpunktes der j -Bézierkurve
bezY[i][j]	y-Koordinate des i -ten Kontrollpunktes der j -Bézierkurve
bezN	Aktuelle Anzahl der generierten Bézierkurven
remX[i]	x-Koordinate des i -ten Punktes im aktuellen Rest-B-Splines
remY[i]	y-Koordinate des i -ten Punktes im aktuellen Rest-B-Splines
remT[i]	i -ter Knoten des aktuellen Rest-B-Splines

Ergebnisse



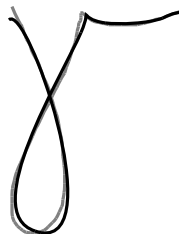
`maxAlteration = 0.2`

(60 Punkte entfernt)



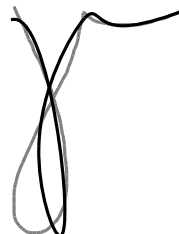
`maxAlteration = 0.5`

(79 Punkte entfernt)



`maxAlteration = 1`

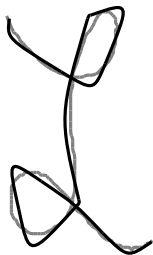
(84 Punkte entfernt)



`maxAlteration = 2`

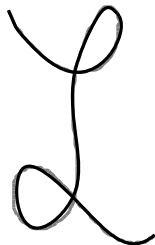
(85 Punkte entfernt)

Vergleich von vier Kurven mit `order = 4`, `windowCapacity = 5`, `useNormL2 = false` und `useEventTime = true` für verschiedene Werte von `maxAlteration`. Die ursprüngliche Kurve (grau) besteht aus 93 Punkten.



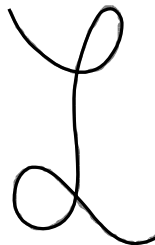
order = 3

(139 Punkte entfernt)



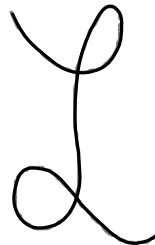
order = 4

(137 Punkte entfernt)



order = 5

(132 Punkte entfernt)



order = 6

(106 Punkte entfernt)

Vergleich von vier Kurven mit `maxAlteration = 0.5`, `windowCapacity = 5`, `useNormL2 = false` und `useEventTime = true` für verschiedene Werte von `order`. Die ursprüngliche Kurve (grau) besteht aus 151 Punkten.



maxAlteration=0.4

(90 Punkte entfernt)



maxAlteration=0.6

(99 Punkte entfernt)



maxAlteration=0.8

(105 Punkte entfernt)



maxAlteration=1

(107 Punkte entfernt)



maxAlteration=0.4

(110 Punkte entfernt)



maxAlteration=0.6

(111 Punkte entfernt)



maxAlteration=0.8

(113 Punkte entfernt)



maxAlteration=1

(113 Punkte entfernt)

Vergleich von vier Kurven mit `windowCapacity = 5`, `order = 4` und `useEventTime = true` für verschiedene Werte von `maxVariation` und `useNormL2`. Für die oberen vier Kurven gilt `useNormL2 = true`, für die unteren `useNormL2 = false`. Die ursprüngliche Kurve besteht aus 121 Punkten und ist grau eingezeichnet worden.



`useEventTime = false`

(106 Punkte entfernt)

`useEventTime = true`

(112 Punkte entfernt)

Vergleich zweier Kurven mit `maxAlteration = 2`, `windowCapacity = 5`, `order = 4` und `useNormL2 = true`. Die Knoten der linken Kurve wurden mit der Chord-Length-Parametrisierung berechnet (`useEventTime = false`), die der rechten mit der Event-Time-Parametrisierung (`useEventTime = true`). Die ursprüngliche Kurve besteht aus 121 Punkten und ist grau eingezeichnet worden. Beim Zeichnen dieser Kurve wurde der Stift eine Sekunde lang an der Spitze der Kurve gehalten.