








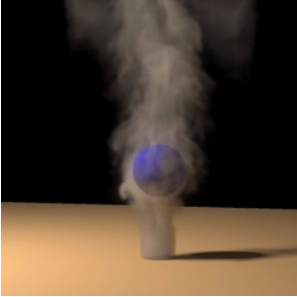
# Virtuelle Realität Keyframe Animations

G. Zachmann  
Clausthal University, Germany  
[cg.in.tu-clausthal.de](http://cg.in.tu-clausthal.de)



## Animationstechniken

- Procedural Animation
  - Particle Systems, Flocks, Crowds, Cloth, Fire, Smoke, Water, ...
- Motion capture
  - Filtering, editing, retargeting, stitching,...



G. Zachmann Virtuelle Realität und Simulation – WS 09/10

Keyframe-Animation 2

■ Scripted Animation
 

- Keyframe interpolation, articulated models, inverse kinematics, deformations, ...

Luxo Jr. movie

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 3

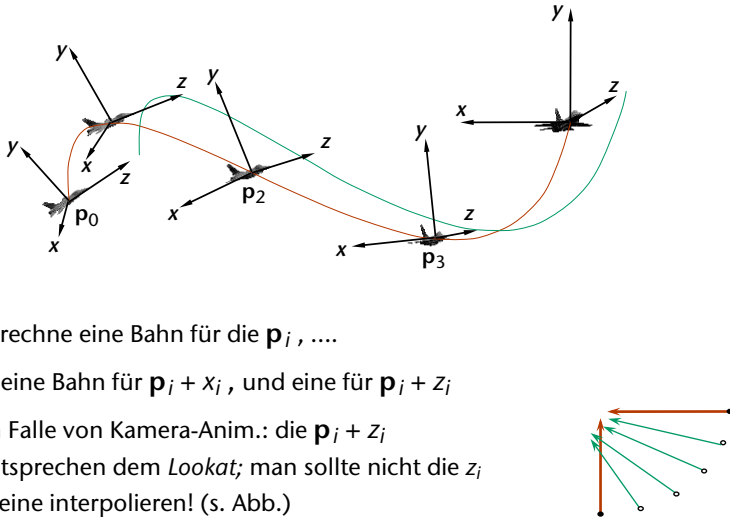
■ Keyframe Animation
 

- Gegeben: die Position / Orientierung  $\mathbf{p}_i$  eines Objektes zu verschiedenen Zeitpunkten  $t_i$ 
  - Funktioniert genauso für die Kamera
- Aufgabe: zu jedem beliebigen Zeitpunkt  $t$  eine Position / Orientierung  $\mathbf{p}$  "dazwischen" berechnen

 → Animation = Interpolation

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 4

▪ Reduktion auf Berechnung von Positionen:

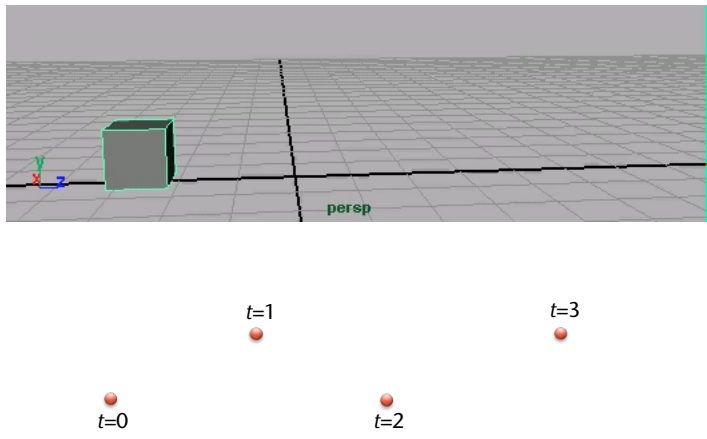


- Berechne eine Bahn für die  $\mathbf{p}_i$ , ....
- ... eine Bahn für  $\mathbf{p}_i + \mathbf{x}_i$ , und eine für  $\mathbf{p}_i + \mathbf{z}_i$
- Im Falle von Kamera-Anim.: die  $\mathbf{p}_i + \mathbf{z}_i$  entsprechen dem *Lookat*; man sollte nicht die  $\mathbf{z}_i$  alleine interpolieren! (s. Abb.)

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 5

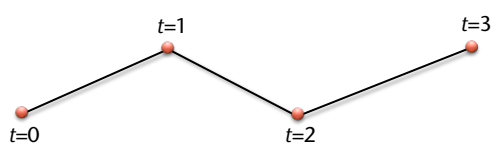
Zur Stetigkeit

- Ein Beispiel: die einzelnen Keyframes

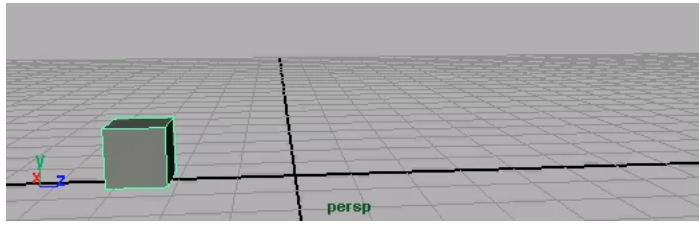


G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 6

■ Lineare Interpolation:



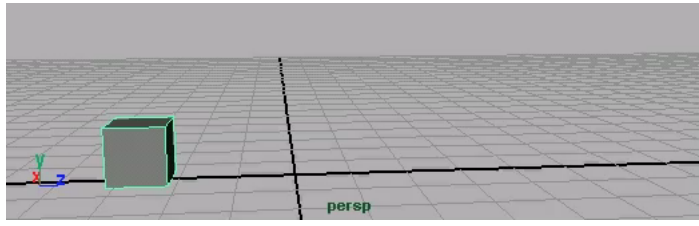
The diagram shows four red dots representing keyframes at time steps  $t=0$ ,  $t=1$ ,  $t=2$ , and  $t=3$ . Straight black lines connect  $t=0$  to  $t=1$ ,  $t=1$  to  $t=2$ , and  $t=2$  to  $t=3$ , illustrating linear interpolation between keyframes.



The 3D scene shows a gray cube on a gray grid floor in a perspective view. A coordinate system with x, y, and z axes is visible in the bottom left corner. The word "persp" is written in the bottom right corner of the scene.

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 7

■ Smooth Interpolation:



The 3D scene shows a gray cube on a gray grid floor in a perspective view. A coordinate system with x, y, and z axes is visible in the bottom left corner. The word "persp" is written in the bottom right corner of the scene.

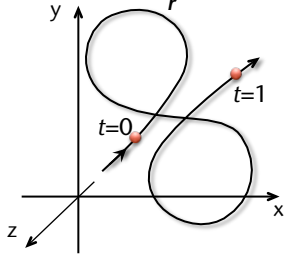
■ In der Animationsgemeinde heißt Interpolation manchmal auch "tweening" (von "in-between")

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 8

## Parametrische Kurven

- Alg. Form:
 
$$\mathbf{r}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}, t \in \mathbb{R}$$
- Ableitung:
 
$$\mathbf{r}'(t) = \begin{pmatrix} x'(t) \\ y'(t) \\ z'(t) \end{pmatrix}$$
- Polynomielle parametrische Kurve:  $\mathbf{r}(t) = \sum_{l=0}^n \mathbf{c}_l t^l$
- Kubische polynomielle Kurve:
 
$$\mathbf{r}(t) = \mathbf{c}_0 + \mathbf{c}_1 t + \mathbf{c}_2 t^2 + \mathbf{c}_3 t^3 = \mathbf{C} \mathbf{u}$$

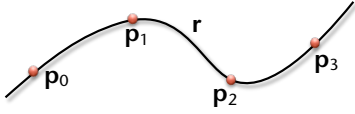
$$\mathbf{C} = \begin{pmatrix} c_{0x} & \cdots & c_{3x} \\ c_{0y} & \cdots & c_{3y} \\ c_{0z} & \cdots & c_{3z} \end{pmatrix} = (\mathbf{c}_0 \quad \mathbf{c}_1 \quad \mathbf{c}_2 \quad \mathbf{c}_3), \quad \mathbf{u} = \mathbf{u}(t) = \begin{pmatrix} 1 \\ t \\ t^2 \\ t^3 \end{pmatrix}$$



G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 9

## Polynomielle kubische Interpolation

- Gegeben: 4 Kontrollpunkte (Datenpunkte)
 
$$\mathbf{p}_i = \begin{pmatrix} x_i \\ y_i \\ z_i \end{pmatrix}, i = 0 \dots 3$$

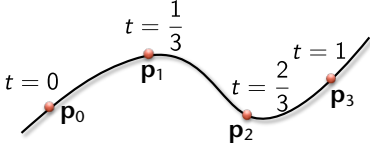


- Gesucht: Koeffizienten  $\mathbf{c}_i$ , so daß die Kontrollpunkte interpoliert werden, d.h.
 
$$\mathbf{r}(t_k) = \mathbf{p}_k \quad \text{für gewisse } t_k$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 10

- **Uniforme polynomielle Interpolation:** verlange, daß

$$\mathbf{r}(0) = \mathbf{p}_0, \mathbf{r}\left(\frac{1}{3}\right) = \mathbf{p}_1$$

$$\mathbf{r}\left(\frac{2}{3}\right) = \mathbf{p}_2, \mathbf{r}(1) = \mathbf{p}_3$$


- Nicht-uniforme Interpolation geht analog

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 11

- Die Interpolationsbedingungen sind spezielle sog. **geometrische Interpolationsbedingungen**
- Hier:

$$\mathbf{r}(0) = \mathbf{c}_0 \quad \stackrel{!}{=} \mathbf{p}_0$$

$$\mathbf{r}\left(\frac{1}{3}\right) = \mathbf{c}_0 + \frac{1}{3}\mathbf{c}_1 + \left(\frac{1}{3}\right)^2 \mathbf{c}_2 + \left(\frac{1}{3}\right)^3 \mathbf{c}_3 \quad \stackrel{!}{=} \mathbf{p}_1$$

$$\mathbf{r}\left(\frac{2}{3}\right) = \mathbf{c}_0 + \frac{2}{3}\mathbf{c}_1 + \left(\frac{2}{3}\right)^2 \mathbf{c}_2 + \left(\frac{2}{3}\right)^3 \mathbf{c}_3 \quad \stackrel{!}{=} \mathbf{p}_2$$

$$\mathbf{r}(1) = \mathbf{c}_0 + \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3 \quad \stackrel{!}{=} \mathbf{p}_3$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 12

▪ Umschreiben in Matrix-Form:

$$C \cdot A = P$$

$$C = (\mathbf{c}_0 \quad \mathbf{c}_1 \quad \mathbf{c}_2 \quad \mathbf{c}_3) \quad A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & (\frac{1}{3})^2 & (\frac{2}{3})^2 & 1 \\ 0 & (\frac{1}{3})^3 & (\frac{2}{3})^3 & 1 \end{pmatrix}$$

$$P = (\mathbf{p}_0 \quad \mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3)$$

▪ Daraus C berechnen:

$$M_I = A^{-1} \Rightarrow C = P \cdot M_I$$

▪ Zusammen:

$$\mathbf{r}(t) = P \cdot M_I \cdot \mathbf{u}$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 13

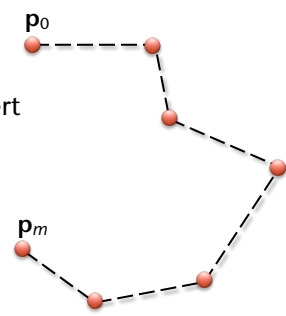
Interpolation über eine Sequenz von Kontrollpunkten

▪ Gegeben:  $\mathbf{p}_0, \dots, \mathbf{p}_m \in \mathbb{R}^3$

▪ Gesucht: eine Kurve, die alle  $\mathbf{p}_i$  interpoliert

▪ Alternative 1: polynomielle Kurve vom Grad  $m+1$  erstellen

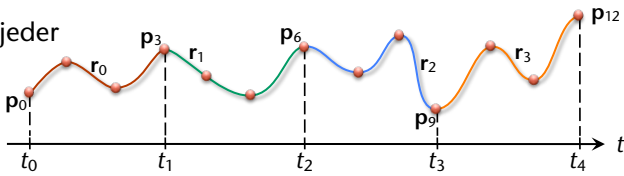
- Matrix-Inversion der Größe  $(m+1) \times (m+1)$  kostet  $O(m^3)$
- Kurve kann sehr stark ausschlagen
- Änderung *eines* Kontrollpunktes hat Auswirkungen auf die *ganze* Kurve! (keine sog. "local control")



G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 14

### Alternative 2

- Zerlege die Folge der  $\mathbf{p}_i$  in 4-er Gruppen,
- Konstruiere zu jeder das interpol. Polynom



- Bestimme zu  $t \in \mathbb{R}$  das Intervall  $[t_i, t_{i+1}]$
- Bilde dann  $t \rightarrow t' \in [0, 1]$  ab
- Berechne  $\mathbf{r}(t) := \mathbf{r}_i(t')$
- Erinnerung: die Matrix  $M_I$  ist für alle 4-er Gruppen gleich!
- Vorteil: **local control**
- Nachteil:  $\mathbf{r}'$  ist an den Nahtstellen ( $\mathbf{p}_3, \mathbf{p}_6, \mathbf{p}_9, \dots$ ) nicht definiert!  
 $\mathbf{r}$  ist nur  $C^0$ -stetig

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 15

### Blending-Funktionen

- Durch Umschreiben der Kurve kann man weitere Einsichten erlangen:

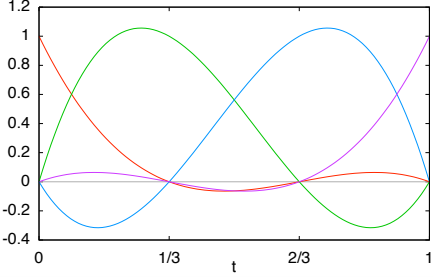
$$\mathbf{r}(t) = C \cdot \mathbf{u}(t) = P \cdot \underbrace{M_I}_{\mathbf{b}(t)} \cdot \mathbf{u}(t) = P \cdot \mathbf{b}(t)$$

$$\mathbf{b}(t) = \begin{pmatrix} b_0(t) \\ \vdots \\ b_3(t) \end{pmatrix}$$

- M.a.W.:

$$\mathbf{r}(t) = \sum_{i=0}^3 \mathbf{b}_i(t) \mathbf{p}_i$$

- Diese  $\mathbf{b}_i$  "mischen" also die Kontrollpunkte, daher heißen sie **blending functions**

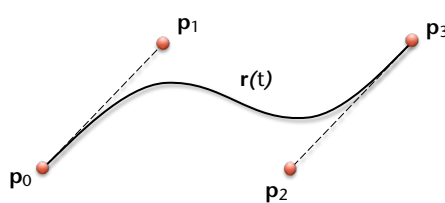


G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 16



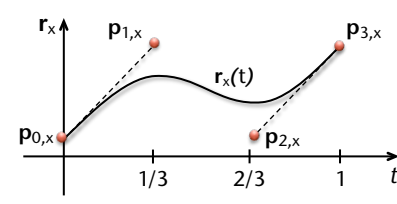
## Kubische Bézier-Kurven

- Ausgangspunkt: große Kurve wird dargestellt durch stückweise polynomielle Kurve
- Langfristiges Ziel: Höhere Stetigkeit an den Anschlußstellen
- Idee: gebe Ableitung in den Punkten  $r(0)$  und  $r(1)$  vor
- Intuitive Art der Vorgabe:



G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 17

- Annahme ab jetzt: **uniforme** Bézier-Kurve (s. Abb.)
  - Analog funktioniert nicht-uniforme
- Betrachte zunächst nur die x-Komponente der Kurve:



- Ansatz ist wieder:

$$r(t) = c_0 + c_1 t + c_2 t^2 + c_3 t^3$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 18

- Geometrische Bedingungen an eine Bézier-Kurve :
 
$$\mathbf{r}(0) \stackrel{!}{=} \mathbf{p}_0 \qquad \mathbf{r}(1) \stackrel{!}{=} \mathbf{p}_3$$

$$\mathbf{r}'(0) \stackrel{!}{=} \frac{\mathbf{p}_1 - \mathbf{p}_0}{\frac{1}{3}} = 3(\mathbf{p}_1 - \mathbf{p}_0) \qquad \mathbf{r}'(1) \stackrel{!}{=} \frac{\mathbf{p}_3 - \mathbf{p}_2}{1 - \frac{2}{3}} = 3(\mathbf{p}_3 - \mathbf{p}_2)$$
- Ableitung von  $\mathbf{r}$  :
 
$$\mathbf{r}'(t) = \mathbf{c}_1 + 2\mathbf{c}_2t + 3\mathbf{c}_3t^2$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 19

- Einsetzen liefert:
 
$$\mathbf{c}_0 = \mathbf{p}_0$$

$$\mathbf{c}_0 + \mathbf{c}_1 + \mathbf{c}_2 + \mathbf{c}_3 = \mathbf{p}_3$$

$$\mathbf{c}_1 = 3\mathbf{p}_1 - 3\mathbf{p}_0$$

$$\mathbf{c}_1 + 2\mathbf{c}_2 + 3\mathbf{c}_3 = 3\mathbf{p}_3 - 3\mathbf{p}_2$$
- Ergibt Gleichungssystem:
 
$$C \cdot A = P$$

$$C = (\mathbf{c}_0 \quad \mathbf{c}_1 \quad \mathbf{c}_2 \quad \mathbf{c}_3) \quad A = \begin{pmatrix} 1 & 1 & 1 & 1 \\ 0 & \frac{1}{3} & \frac{2}{3} & 1 \\ 0 & 0 & \frac{1}{3} & 1 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$P = (\mathbf{p}_0 \quad \mathbf{p}_1 \quad \mathbf{p}_2 \quad \mathbf{p}_3)$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 20

- Daraus die Bézier-Geometriematrix berechnen:

$$M_B = A^{-1} = \begin{pmatrix} 1 & -3 & 3 & -1 \\ 0 & 3 & -6 & 3 \\ 0 & 0 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

- Die Blending-Funktionen sind somit:

$$\mathbf{r}(t) = P \cdot \underbrace{M_B}_{\mathbf{b}(t)} \cdot \mathbf{u}(t) = P \cdot \mathbf{b}(t), \quad \mathbf{b}(t) = \begin{pmatrix} (1-t)^3 \\ 3t(1-t)^2 \\ 3t^2(1-t) \\ t^3 \end{pmatrix}$$

- Diese heißen **Benstein-Polynome**

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 21

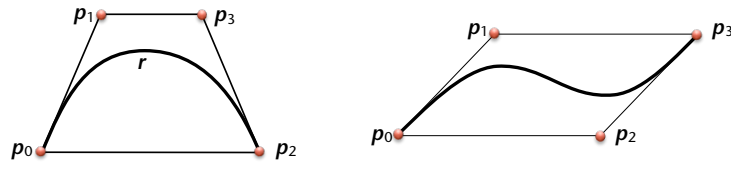
### Eigenschaften der Bernstein-Polynome

- Alle Nullstellen liegen bei  $t=0$  oder  $t=1$
- $\forall t \in [0, 1] : 0 \leq b_i(t) \leq 1$
- Es gilt die "Teilung der 1" (*partition of unity*):

$$\sum_{i=0}^3 b_i(t) \equiv 1$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 22

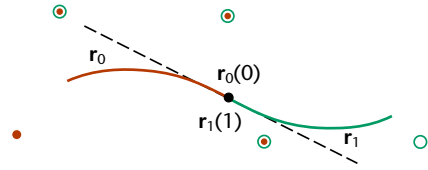
- Folge:  $r(t)$  ist eine *baryzentrische* Kombination der  $p_i$
- Folge daraus ist eine wichtige Eigenschaft der Bézier-Kurve:  
"convex hull property"!  
D.h.,  $r(t)$  liegt für alle  $t \in [0, 1]$  innerhalb der konvexen Hülle der  $p_i$  !



G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 23

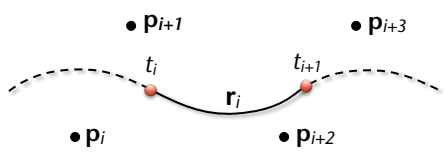
## B-Spline-Kurven

- Problem: Kurve durch  $n$  Kontrollpunkte festlegen
- Alternative 1: Bézierkurve vom Grad  $n-1$ 
  - Interpoliert die Kontrollpunkte
  - Aber immer noch kein *local control* (wenn auch der Einfluß weit entfernter K.punkte schwächer als bei polynomieller Interpolation ist)
- Alternative 2: stückweise Bézierkurve
  - Vorteil: local control
  - Nachteil: nur  $C^0$ -stetig
- Idee:
  - Verzichte auf Interpolation, also nur **Approximation**
  - Fordere dafür an den Ausschlußstellen  $C^1$ -Stetigkeit



G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 24

- Approximiere  $n+1$  Kontrollpunkte  $\mathbf{p}_0, \dots, \mathbf{p}_n$  durch eine Spline-Kurve bestehend aus  $n-2$  Segmenten  $\mathbf{r}_0, \dots, \mathbf{r}_{n-3}$
- Das Segment  $\mathbf{r}_i$  wird definiert durch  $\mathbf{p}_i, \mathbf{p}_{i+1}, \mathbf{p}_{i+2}, \mathbf{p}_{i+3}$



- Die Ausschlußstellen zwischen  $\mathbf{r}_i$  und den Nachbarsegmenten heißen **Knoten (knots)**
- Das Segment ist definiert über  $t \in [t_i, t_{i+1}]$
- D.h., die ganze Kurve geht über  $t \in [0, n]$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 25

### Vereinfachung: kubische, uniforme B-Splines

- Vereinfachung: **kubische** B-Splines
  - Damit ist
 
$$\mathbf{r}_i(t) = \sum_{j=0}^3 \mathbf{c}_j (t - t_i)^j$$
- Vereinfachung : **uniforme** B-Splines
  - D.h.  $t_{i+1} = t_i + 1$
  - O.B.d.A:  $t_0 = 0$
  - Definiere:  $t' = t - t_i, \quad t' \in [0, 1]$ 

$$\mathbf{r}_i(t) = \mathbf{r}'_i(t') = \sum_{j=0}^3 \mathbf{c}_j t'^j$$
- Schreibe ab jetzt einfach  $\mathbf{r}_i(t) = \sum_{j=0}^3 \mathbf{c}_j t^j$  , wenn Kontext klar

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 26

**Geometrische Bedingungen**

- Wie bisher suchen wir die Polynom-Koeffizienten  $\mathbf{c}_0, \mathbf{c}_1, \mathbf{c}_2, \mathbf{c}_3$
- Betrachte das Segment  $\mathbf{r}_i$

**1. Geometrische Bedingung (für  $C^0$ -Stetigkeit):  $\mathbf{r}_i(1) \stackrel{!}{=} \mathbf{r}_{i+1}(0)$**

- Beobachtung:  $\mathbf{r}_i(1)$  kann nur eine gewichtete Summe der Kontrollpunkte  $\mathbf{p}_i, \dots, \mathbf{p}_{i+3}$  sein
- In  $\mathbf{r}_{i+1}(0)$  können nur  $\mathbf{p}_{i+1}, \dots, \mathbf{p}_{i+4}$  eingehen; und (s.o.)  $\mathbf{r}_i(1) \stackrel{!}{=} \mathbf{r}_{i+1}(0)$
- Fazit:  $\mathbf{r}_i(1)$  kann nur gewichtete Summe von  $\mathbf{p}_{i+1}, \mathbf{p}_{i+2}, \mathbf{p}_{i+3}$  sein
- Sinnvoll erscheint außerdem, dass  $\mathbf{p}_{i+2}$  (mittlerer Knoten) höher gewichtet wird als  $\mathbf{p}_{i+1}$  und  $\mathbf{p}_{i+3}$
- Bei den B-Splines wählt man

$$\mathbf{r}_i(1) = \mathbf{r}_{i+1}(0) = \frac{1}{6}(\mathbf{p}_{i+1} + 4\mathbf{p}_{i+2} + \mathbf{p}_{i+3})$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 27

**2. Geometrische Bedingung (für  $C^1$ -Stetigkeit):**

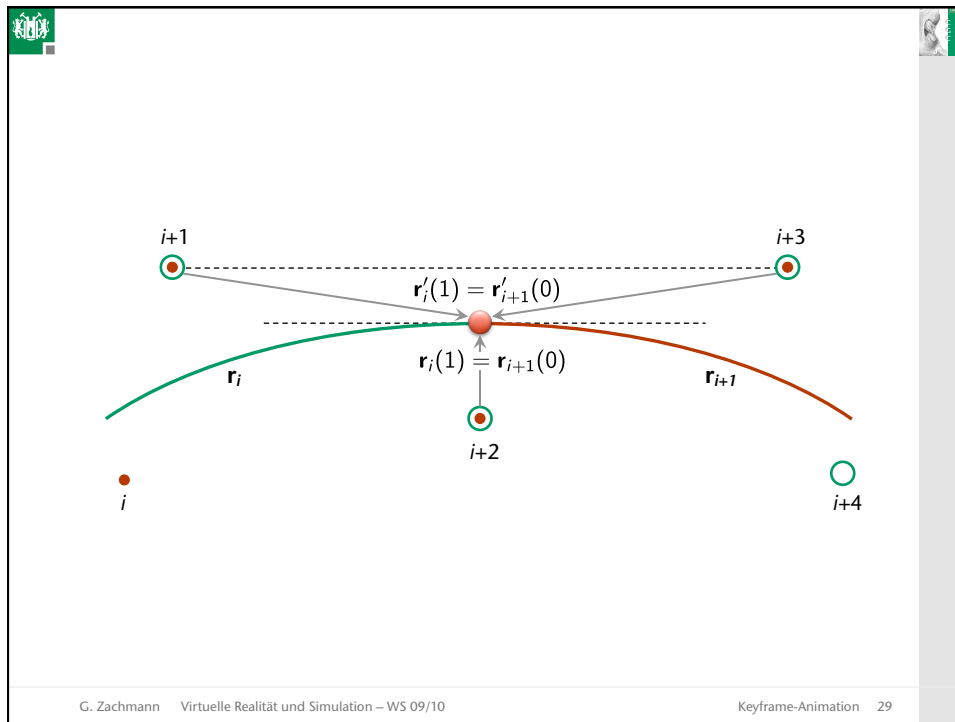
$$\mathbf{r}'_i(1) \stackrel{!}{=} \mathbf{r}'_{i+1}(0)$$

- Selbe Beobachtungen wie oben  $\rightarrow$  nur  $i+1, i+2, i+3$  können eingeben
- Sinnvoll erscheint,  $\mathbf{r}'_i(1)$  an der "Steigung" dieser 3 Kontrollpunkte zu orientieren
- Bei den B-Splines wählt man:

$$\mathbf{r}'_i(1) = \frac{1}{2}(\mathbf{p}_{i+3} - \mathbf{p}_{i+1})$$

- Bemerkung zum Faktor  $\frac{1}{2}$ : die Punkte  $\mathbf{p}_{i+1}$  und  $\mathbf{p}_{i+3}$  "überstreichen" im Schnitt ein Parameter-Intervall  $\Delta t = 2$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 28



### 3. und 4. Geometrische Bedingung:

- Wünschenswert: symmetrische Kurve bzgl. der Nummerierung
- Die Bedingungen an  $\mathbf{r}_i(0)$  und  $\mathbf{r}'_i(0)$  können, mit demselben Argument wie oben, nur die Punkte  $\mathbf{p}_i$ ,  $\mathbf{p}_{i+1}$ ,  $\mathbf{p}_{i+2}$  einbeziehen (wegen  $\mathbf{r}_{i-1}$ ).

▪ Fazit: 
$$\mathbf{r}_i(0) = \frac{1}{6} (\mathbf{p}_i + 4\mathbf{p}_{i+1} + \mathbf{p}_{i+2})$$

$$\mathbf{r}'_i(0) = \frac{1}{2} (\mathbf{p}_{i+2} - \mathbf{p}_i)$$

- 4 Bedingungen für 4 Unbekannte; LGS aufstellen; Resultat:

$$\mathbf{r}_i(t) = P \cdot M_S \cdot \mathbf{u}(t), \quad M_S = \begin{pmatrix} 1 & -3 & 3 & -1 \\ 4 & 0 & -6 & 3 \\ 1 & 3 & 3 & -3 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

▪ Die Blending-Polynome:

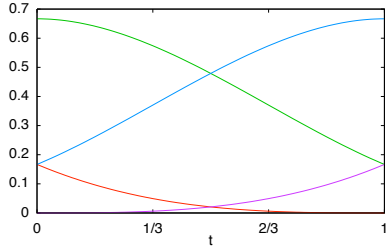
$$\mathbf{r}_i(t) = P \cdot \mathbf{b}(t)$$

$$\mathbf{b}(t) = M_S \cdot \mathbf{u}(t) = \frac{1}{6} \begin{pmatrix} (1-t)^3 \\ 4-6t^2+3t^3 \\ 1+3t+3t^2-3t^3 \\ t^3 \end{pmatrix} = \begin{pmatrix} b_0(t) \\ b_1(t) \\ b_2(t) \\ b_3(t) \end{pmatrix}$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 32

Eigenschaften der B-Splines

- $\forall t \in [0, 1] : 0 \leq b_i(t) \leq 1$
- Teilung der 1:
 
$$\sum_{i=0}^3 b_i(t) = 1$$
- Convex Hull Property
- Die Blending-Funktionen sind für alle  $\mathbf{r}_i$  gleich!
- Invariant unter affinen Transformationen:
 
$$\Phi(\mathbf{p}(t; p_0, \dots, p_n)) = \mathbf{p}(t; \Phi(p_0), \dots, \Phi(p_n))$$
- Stetigkeit:
 
$$\mathbf{p}(t) \in \mathcal{C}^2$$



G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 33



- Auswertung:
  - Gegeben  $t \in [0, n]$
  - Setze  $i := \lfloor t \rfloor$ ,  $t' = t - i$
  - Naiv ausrechnen:
 
$$\mathbf{p}(t) = \sum_{j=0}^3 b_j(t') \mathbf{p}_{i+1}$$
- Wie bekommt man einen exakten Start- und Endpunkt?
  - M.a.W.: wie kann man erreichen, dass Start- und Endpunkt doch interpoliert werden?
  - Trick: Start- / Endpunkt 3 mal wiederholen

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 34

### Die B-Spline-Basis

- Wie viele Segmente eines B-Splines beeinflusst ein Kontrollpunkt  $\mathbf{p}_i$ ,  $3 \leq i \leq n-3$  ?
- Genau 4 aufeinanderfolgende:
 

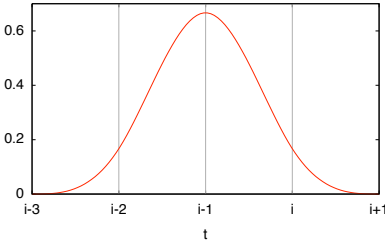
$t$	$\mathbf{p}'s$	Blending-Fkt.
$[i, i + 1]$	$i, i + 1, i + 2, i + 3$	$b_0$
$[i - 1, i]$	$i - 1, i, i + 1, i + 2$	$b_1$
$[i - 2, i - 1]$	$i - 2, i - 1, i, i + 1$	$b_2$
$[i - 3, i - 2]$	$i - 3, i - 2, i - 1, i$	$b_3$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 35

- Repräsentiere den Einfluß eines Punktes als Funktion:

$$B_i(t) = \begin{cases} 0 & t < i-3 \\ b_3(t-i+3) & i-3 \leq t < i-2 \\ b_2(t-i+2) & i-2 \leq t < i-1 \\ b_1(t-i+1) & i-1 \leq t < i \\ b_0(t-i) & i \leq t < i+1 \\ 0 & t \geq i+1 \end{cases}$$

- Diese  $B_i$  heißen **Basis-Funktionen**



G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 36

- Damit kann man dann den gesamten B-Spline auch als eine Summe schreiben:

$$\mathbf{p}(t) = \sum_{i=3}^{n-3} B_i(t) \mathbf{p}_i$$


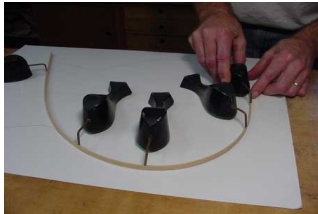
- Nebenbemerkung: die  $B_i'$  sind nur verschiedene Varianten derselben Funktion:

$$B_i(t) = B_{i-1}(t+1)$$

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 37

## Geschichte der Splines

- Vor dem Computer: Ingenieure verwendeten Gewichte ("Ducks") und dünne Holzplanken ("splines") zur Konstruktion von glatten Kurven
- Ergibt Kurven, die
  - $C^2$ -stetig sind (mindestens)
  - durch die Kontrollpunkte verlaufen (interpolierend sind)

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 38

## Demo zur polynomiellen Interpolation

index.html

file:///Users/zach/Data/Demos/CAGD/Interpolation/index.html

index.html

### DotPlacer Applet

back  Remove  Move  Add  Animate

<input checked="" type="checkbox"/>	axes
<input checked="" type="checkbox"/>	grid
<input checked="" type="checkbox"/>	dots
<input checked="" type="checkbox"/>	blobs
<input checked="" type="checkbox"/>	circles
<input checked="" type="checkbox"/>	x's
<input checked="" type="checkbox"/>	+s
<input checked="" type="checkbox"/>	voronoi
<input checked="" type="checkbox"/>	delauney
<input checked="" type="checkbox"/>	hull
<input checked="" type="checkbox"/>	bezier
<input checked="" type="checkbox"/>	lines
<input checked="" type="checkbox"/>	spline
<input checked="" type="checkbox"/>	lin int
<input checked="" type="checkbox"/>	poly int
<input checked="" type="checkbox"/>	1.sq. line
<input checked="" type="checkbox"/>	1.sq. quad
<input checked="" type="checkbox"/>	1.sq. cubic
<input checked="" type="checkbox"/>	1.sq. quartic
<input checked="" type="checkbox"/>	copyright

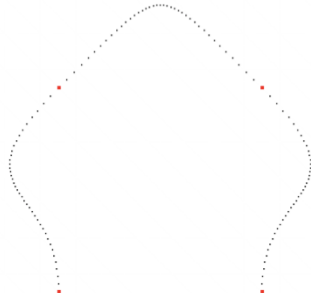
This Applet illegally copied or linked.  
Original is at: <http://www.dr-mikes-maths.com/DotPlacer.html>

This applet helps you visualize a number of mathematical constructions. They are explained one by one below. They are each constructed from a set of points in the plane. Initially, the drawing plane is empty.

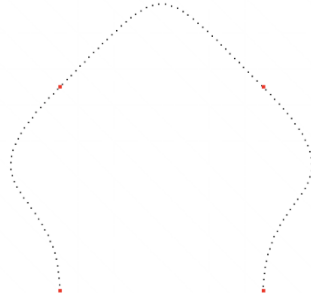
G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 39

## Erstes Problem: gleichmäßiges Sampling der Kurve

- Beispiel:
 



Uniforme  
Parameterintervalle

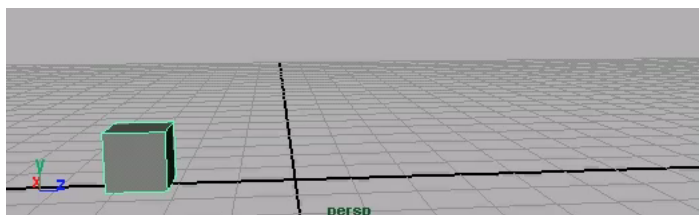


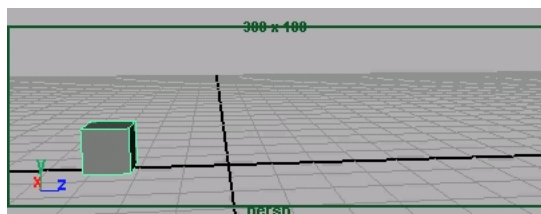
Konstante  
Geschwindigkeit
- Fazit: konstante Geschwindigkeit bekommt man *nicht* durch uniformes Sampling! ☹

G. Zachmann Virtuelle Realität und Simulation – WS 09/10
Keyframe-Animation 40

## Motion Curves

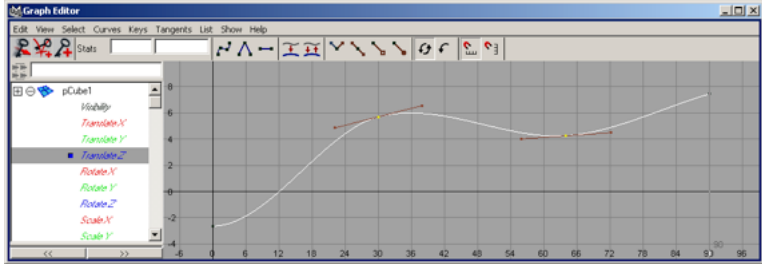
- Wiederholung – diese Interpolation ist smooth:
 


- Auch glatt, aber hier bekommt man den Eindruck eines bestimmten Gewichtes:
 



G. Zachmann Virtuelle Realität und Simulation – WS 09/10
Keyframe-Animation 41

- Der Unterschied: **ease in/out**, or **slow in/out**, means to leave the key slowly
- Führt zu sog. "**motion curves**" = Weg-Zeit-Kurve:



- Zweites Problem: solche Weg-Zeit-Kurven abbilden
  - Ist eigentlich nur eine Verallgemeinerung des ersten Problems

Zur Unterhaltung siehe die "Cartoon Laws of Physics"  
 ([zach.in.tu-clausthal.de/teaching/vr\\_literatur/Cartoon\\_Laws\\_of\\_Physics.html](http://zach.in.tu-clausthal.de/teaching/vr_literatur/Cartoon_Laws_of_Physics.html))

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 42

## Die Parametrisierung

- Gegeben:
 

Parametrische Kurve	$\mathbf{p}(u)$
Weg-Zeit-Kurve (auch param.)	$r(t) : [0, 1] \mapsto [0, 1]$
Zeitpunkt	$t$
- Gesucht: Parameter  $u$ , so daß der zurückgelegte Weg auf der Kurve
 
$$s(u) = \int_0^u |\mathbf{p}'(u)| du = r(t)$$
  - $s(u)$  heißt **Bogenlänge (arc length)**
- Analytische (theoretische) Lösung:
  - $s$  invertieren
  - $s^{-1}(r(t))$  ausrechnen

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 43

## Bogenlängentabelle (Arc Length Table)

- Praktische Lösung:
  - Unterteile Pfad in viele kleine Segmente mit konstanten  $u$ -Intervallen
  - Längen der Segmente aufakkumulieren und in Tabelle speichern
  - Für Zeitpunkt  $t$  berechne  $r(t)$
  - Suche Einträge in Tabelle mit binärer Suche
  - Interpoliere dazwischen  $\rightarrow$  Parameter  $u$
  - Berechne  $s(u)$

$u$	Seg- ment	Bogen- länge (s)
0,000	0	0,00
0,125	0	2,84
0,250	0	5,13
0,375	0	7,09
0,500	0	8,80
0,625	0	10,32
0,750	0	11,98
0,875	0	14,40
1,000	0	18,16
0,125	1	22,16
0,250	1	25,10
0,375	1	27,15
0,500	1	28,62
0,625	1	30,09
0,750	1	32,14
0,875	1	35,07
1,000	1	39,08
0,125	2	42,84
0,250	2	45,26
0,375	2	46,91
0,500	2	48,43
0,625	2	50,15
0,750	2	52,11
0,875	2	54,40
1,000	2	57,24

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 44

## Die Bestandteile einer Keyframe-Animation

1. Keys setzen
  - Einige Positionen auf dem Pfad des Objektes
2. Interpolationsart festlegen
3. Zeit-Weg-Verlauf festlegen
  - Typischerweise wiederum durch Interpolation von 2D "Key"-Punkten
4. Pfad an sehr vielen Stellen interpolieren

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 45

The image is a collage illustrating keyframe animation. It features several distinct scenes and sequences:

- Top Left:** A 3D rendered scene of a desk with a lamp, a red Christmas ornament, and a green branch with red ornaments.
- Top Right:** A sequence of keyframes showing a character's legs in various poses, illustrating the animation of movement.
- Middle Left:** A 3D scene with a man painting a wall, a chair, and a cat.
- Middle Right:** A 3D scene with a chair, a cat, and a desk.
- Bottom Left:** A 3D scene with a tiger.
- Bottom Right:** A sequence of keyframes showing a tiger in various poses, illustrating the animation of movement.

G. Zachmann Virtuelle Realität und Simulation – WS 09/10 Keyframe-Animation 46