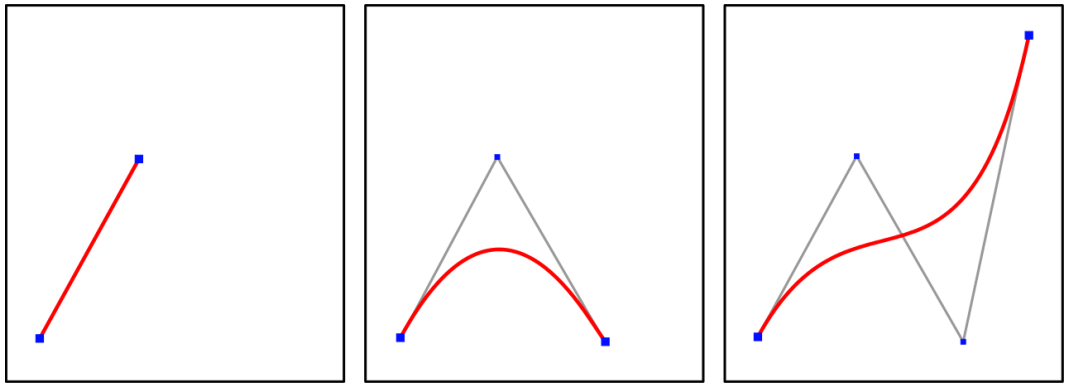
Bézierkurven

- Bézierkurven verwenden Bernstein-Polynome n-ten Grades zur Approximation von $(n+1)$ Stützpunkten $b_0, \dots, b_n \in \mathbb{R}^p$.
- Die durch diese Stützpunkte definierte Kurve

$$p(t) = \sum_{i=0}^{n} b_i \cdot B_i^{(n)}(t)$$

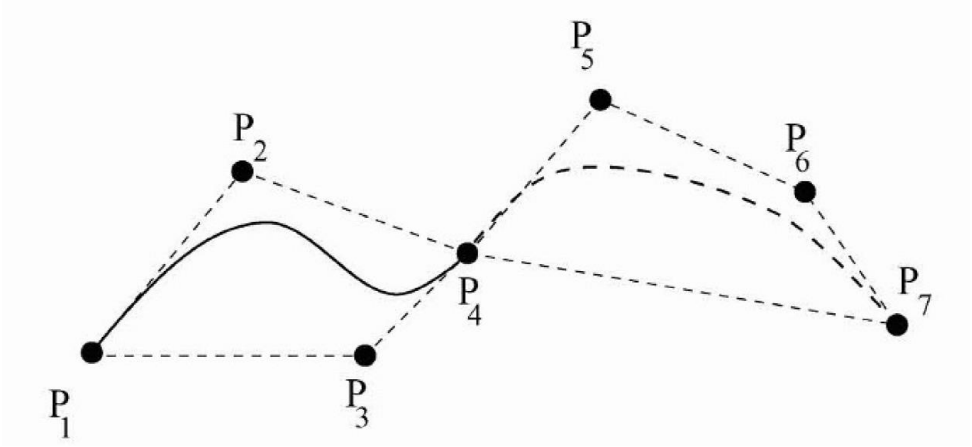
  heißt Bézierkurve vom Grad n.
- Bézierkurven interpolieren den Anfangs- und Endpunkt, die anderen Stützpunkte liegen i.d.R. nicht auf der Kurve.



https://de.wikipedia.org/wiki/B%C3%A9zierkurve

Stückweise Bézierkurven

- Setzen sich aus mehreren Bézierkurven niedrigen Grades n zusammen
  - Üblicherweise dritter oder vierter Grad
- Berechnung einer Bézierkurve für jeweils $n + 1$ aufeinander folgende Stützpunkte
- Letzter Stützpunk der Bézierkurve wird erster Stützpunkt der nächsten usw.
  - Stützpunkte an Nahtstellen werden interpoliert



F. Klawonn: „Grundkurs Computergrafik mit Java“, Vieweg und Teubner, 3. Auflage, 2010

B-Spline-Kurven

- Durch Verzicht auf die Interpolation lässt sich lokale Kontrolle und C2-Kontinuität erreichen
- Möglich durch Austausch der Basisfunktion zur B-Spline-Basisfunktion:

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

B-Spline Basisfunktion:

$$N_{i,0}(t) = \begin{cases} 1 & \text{für} \quad x_i \leq t < x_{i+1} \\ 0 & \text{sonst} \end{cases}$$

$$N_{i,k}(t) = \frac{t - x_i}{x_{i+k} - x_i} N_{i,k-1}(t) + \frac{x_{i+k+1} - t}{x_{i+k+1} - x_{i+1}} N_{i+1,k-1}(t)$$

Cox- de Boor-Rekursion

- Polynom hat den Grad k.
- Knotenvektor aus aufsteigend sortierten Werten (bestimmt durch Anzahl der Stützpunkte n)

$$x_i : x_1 < \cdots < x_n < x_{n+1} < x_{n+k+1}$$

NURBS

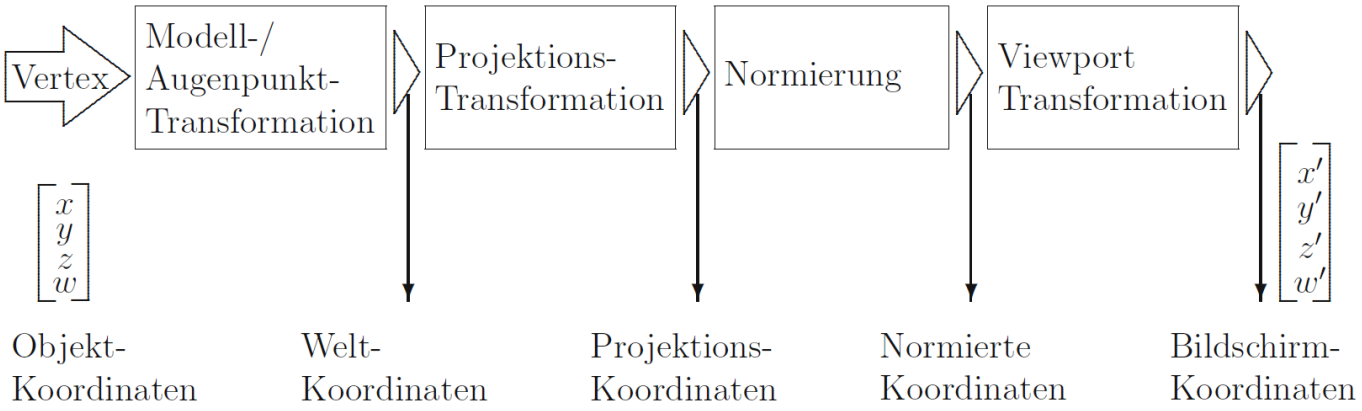
- NURBS = Non-uniform rational B-Splines
  - Erweiterung der Basisfunktionen auf rationale Funktionen
  - Einführung eines Gewichtungsfaktors für jeden Stützpunkt
- Kurve ist definiert durch:

$$p(t) = \frac{1}{\sum_{i=0}^{n-1} N_{i,k}(t) \cdot w_i} \sum_{i=0}^{n-1} b_i \cdot N_{i,k}(t) \cdot w_i$$

$$p(t) = \sum_{i=0}^{n-1} b_i \cdot R_{i,k}(t)$$

mit $R_{i,k}(t) = \frac{N_{i,k}(t) \cdot w_i}{\sum_{j=0}^{n-1} N_{j,k}(t) \cdot w_j}$ NURBS Basisfunktion

Transformationskette



- **Modell-/Augenpunkttransformation**: Positionierung von Objekten in der Szene
  (= ModelView Matrix)
- **Projektionstransformation:** Definition des sichtbaren Volumens, z.B. Blickwinkel
  (= Frustum)
- **Normierung:** Transformation der Koordinaten auf Intervall [-*w*, +*w*] und Division durch inversen Streckungsfaktor *w*
  (= Normalized Device Coordinates)
- **Viewport-Transformation:** Positionierung der Vertices im Fenster

A. Nischwitz et al. „Computergrafik und Bildverarbeitung“, Band 1: Computergrafik“

Transformation mit homogenen Koordinaten

- Allgemein:

$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} m_{11} & m_{12} & m_{13} & m_{14} \\ m_{21} & m_{22} & m_{23} & m_{24} \\ m_{31} & m_{32} & m_{33} & m_{34} \\ m_{41} & m_{42} & m_{43} & m_{44} \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ w \end{pmatrix}$$

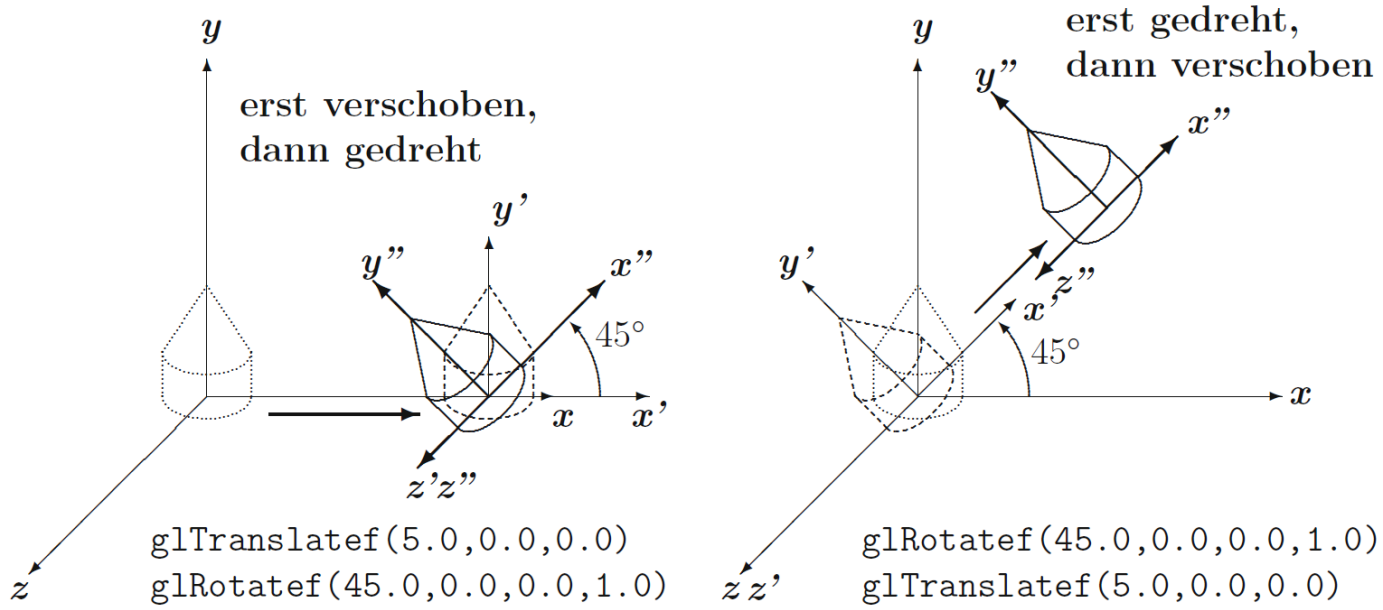
$$v' = \boldsymbol{M} v$$

- Alle Transformationen der Transformationskette arbeiten mit homogenen Koordinaten → Effiziente Abbildung der Operationen in der Hardware
- Ausführung der Transformationen hintereinander
  - Möglichkeit 1: $v' = \left(\ldots \cdot \mathbf{M}_2 \cdot (\mathbf{M}_1 \cdot v)\right)$
  - Möglichkeit 2: $v' = (\ldots \cdot \mathbf{M}_2 \cdot \mathbf{M}_1) \cdot v$

- Affine Transformation (Translation, Rotation, Skalierung) abgebildet in 4x4 Matrizen

Reihenfolge der Transformationen

- Transformation des lokalen Koordinatensystem im Weltkoordinatensystem

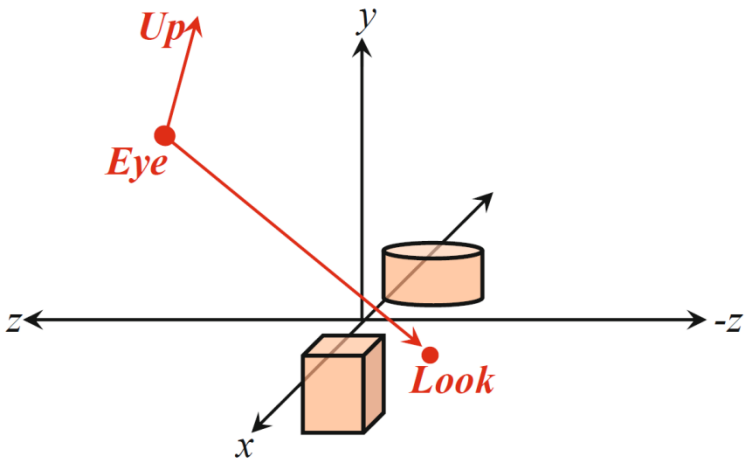


- Im Programmcode: Verkettung durch Rechtsmultiplikation → folgt der Denkweise „Transformation des lokales Koordinatensystem“

A. Nischwitz et al. „Computergrafik und Bildverarbeitung“, Band 1: Computergrafik“

Augpunkttransformationen

- Ändern der Position und der Blickrichtung des Augpunktes
- Gleichbedeutend mit Verschiebung der Szene (Modelltransformation)
- Zusammenfassung in einem Modelview-Matrix-Stack: Augpunkt wird zu $(0,0,0)$ verschoben, Szene entsprechend invers



A. Nischwitz et al. „Computergrafik und Bildverarbeitung“, Band 1: Computergrafik“