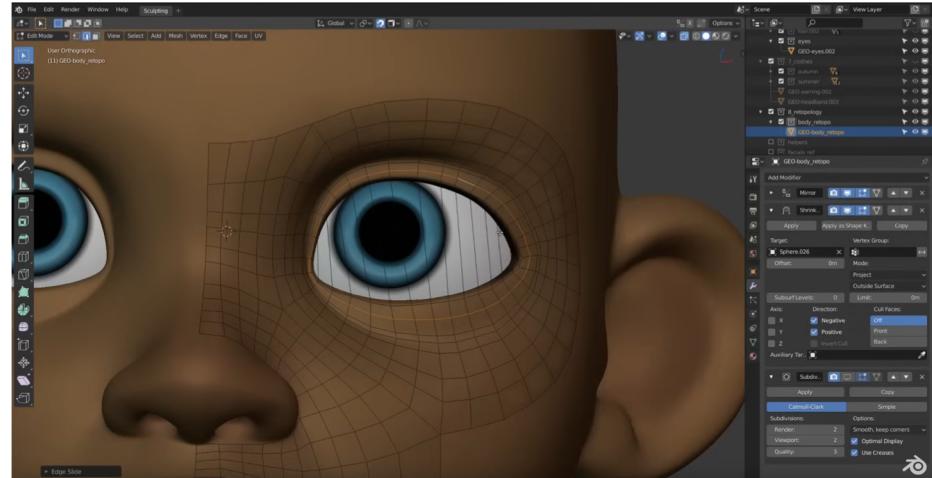


2. Geometrische Modellierung

2. Geometrische Modellierung



- Es gibt verschiedene Möglichkeiten, graphische Daten zu erzeugen. Die klassische Variante ist Modellierung, daneben gibt es auch andere Ansätze:
- 3D Scanner für Geometrie
- Bildrekonstruktion für optische Eigenschaften
- Simulation
 - Strömungsdaten
 - FEM – finite Elemente



3D scannen einer Skulptur

2. Geometrische Modellierung

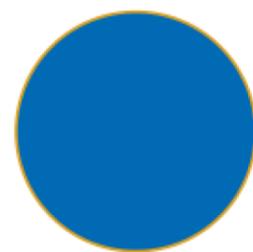
2.1 Begriffs- und Modellbildung

2. Geometrische Modellierung

2.1 Begriffs- und Modellbildung



- Geometrische Objekte sind Objekte der realen oder theoretischen Welt, die durch ihre geometrischen und topologischen Eigenschaften beschrieben sind:
 - Geometrische Eigenschaften beschreiben die präzise Form eines Objekts und die konkrete Lage in einem Bezugssystem.
 - Topologische Eigenschaften definieren die fundamentale Objektstruktur, die bei kontinuierlichen Transformationen invariant ist.



Verschiedene Körper und gleiche (links) bzw. verschiedene Topologie (rechts)

2. Geometrische Modellierung

2.1 Begriffs- und Modellbildung



- **Geometrisches Modellieren ist:**

- rechnergestütztes Entwerfen und Manipulieren von geometrischen Objekten und ihren Lagebeziehungen (CAD) und die

- **Grundlage für:**

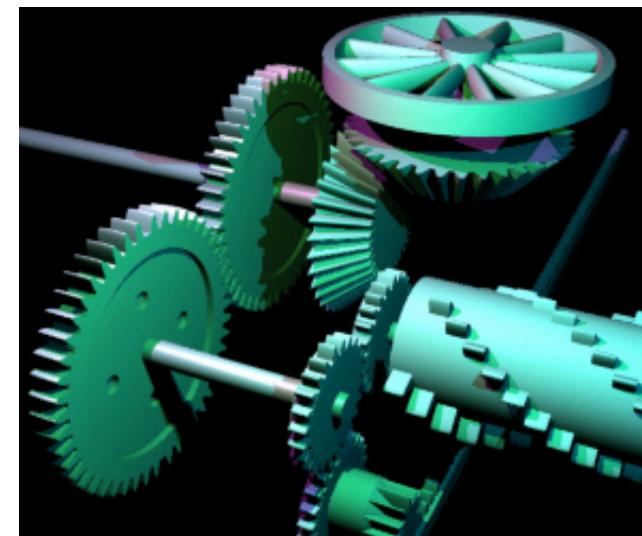
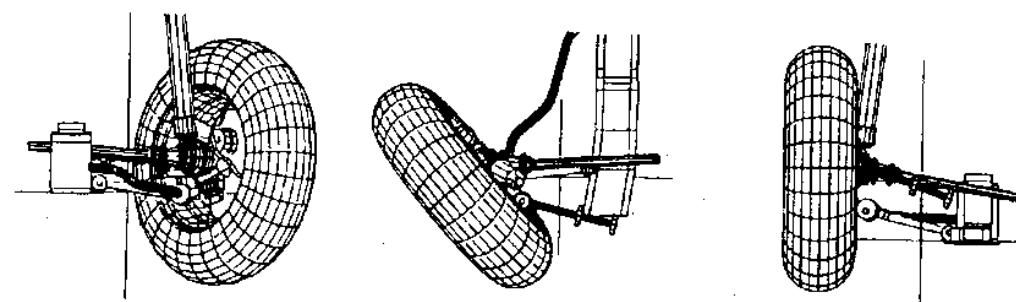
- die Berechnung geometrischer Eigenschaften;
 - die graphische Darstellung der Objekte;
 - die Berechnung des physikalisch-geometrischen Verhaltens der Objekte (bei gegebener Attributierung mit physikalischen Eigenschaften).

2. Geometrische Modellierung

2.1 Begriffs- und Modellbildung



- **3D-Modelle**
- definieren Objekte in 3D durch Abstraktion der wesentlichen geometrischen Eigenschaften.
- Wir unterscheiden 3 verschiedene Modellformen:
 - **Drahtmodelle**
beschreiben Objekte durch Konturlinien;
 - **Flächenmodelle**
beschreiben Objekte durch Randflächen;
 - **Körpermodelle**
beschreiben Objekte als Körper.



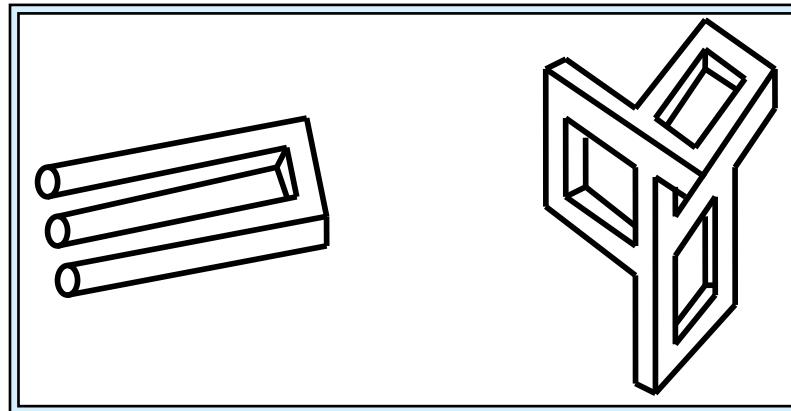


■ **Drahtmodelle (wire frame)**

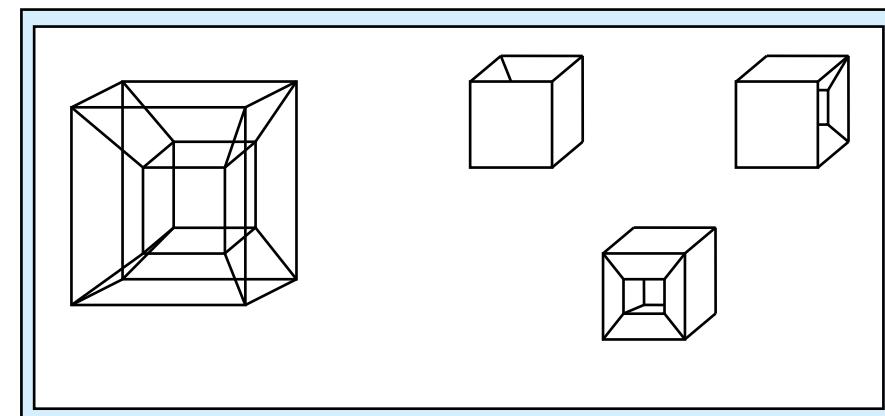
- beschreiben Objekte durch ihre Konturen.
- Zwischen den Elementen zur Beschreibung einer Kontur (Kante, Kreisbögen, Spline, etc.) bestehen keine Beziehungen.
 - Eine Zuordnung von Konturen zu Flächen ist nicht definiert.
 - Drahtmodelle haben folgende Eigenschaften:
 - Sie sind einfach und traditionell.
 - Sie tolerieren Nonsenseobjekte.
 - Unsichtbare Kanten müssen "von Hand" durch Löschen entfernt werden.
 - Schnittflächen lassen sich nicht automatisch erzeugen.
 - Drahtmodelle sind CAD/CAM untauglich.

2. Geometrische Modellierung

2.1 Begriffs- und Modellbildung



*Nonsensobjekte als Drahtmodelle
(Ernst, 1987)*



Mehrdeutigkeit bei Drahtmodellen



■ **Flächenmodelle**

- erweitern das Drahtmodell durch Aufspannen von Flächen zwischen den Konturen. Sie beschreiben Oberflächenstrukturen, aber keine Körper.
- Typische Grundelemente sind Polygone, Oberflächen von Kugeln oder Ellipsen, Freiformflächen u.a.
- Flächen stehen bei einem Flächenmodellierer in keinem gegenseitigen Zusammenhang, d.h. es sind keine Nachbarschaftsbeziehungen abgespeichert.
- Flächenmodelle werden vor allem dort eingesetzt, wo es auf die Eigenschaften und die Form von Flächen ankommt (z.B. im Automobilbau).

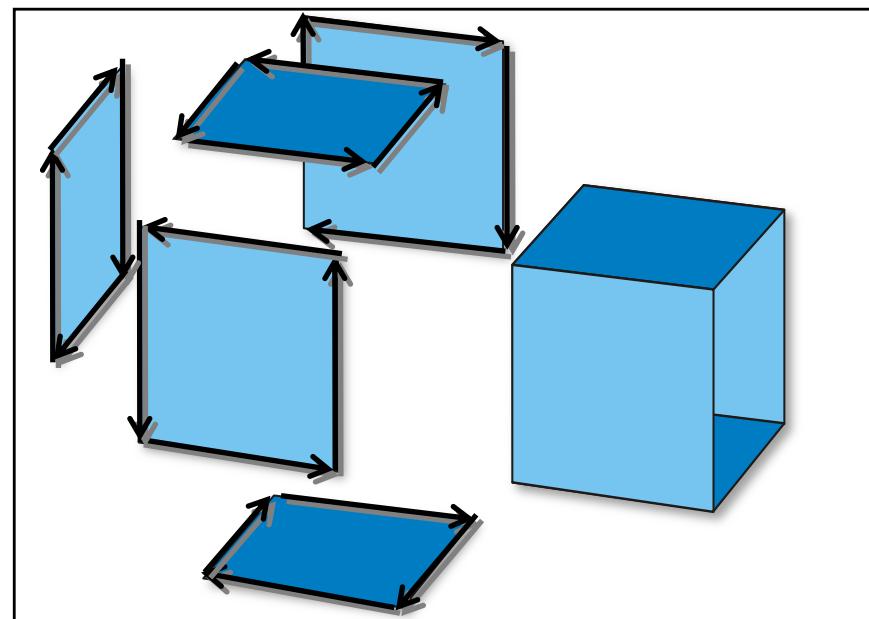
2. Geometrische Modellierung

2.1 Begriffs- und Modellbildung

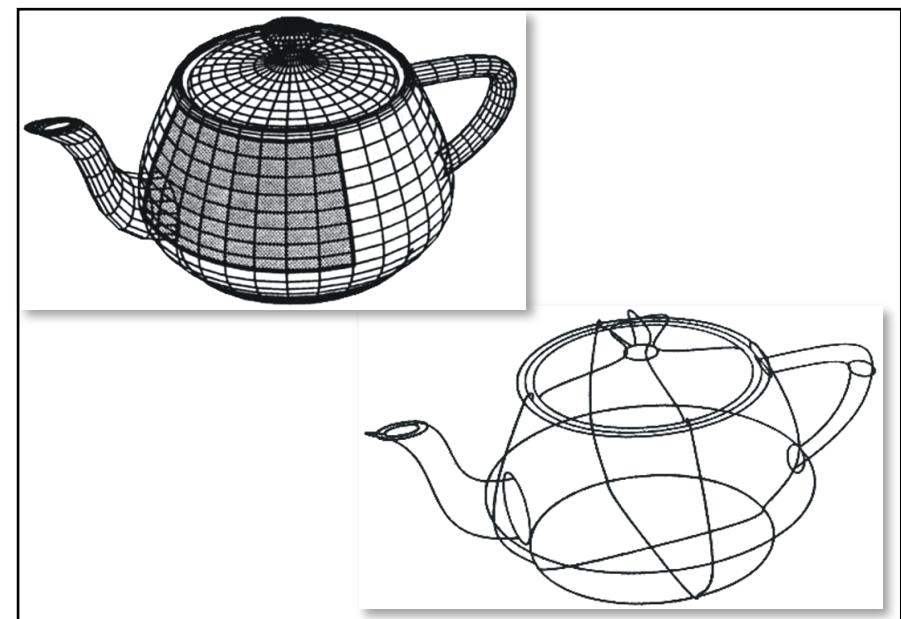


Beispiele für Flächenmodelle:

polygonale Flächenrepräsentation



parametrische Flächenrepräsentation mit
32 Bezier-Patches



aus: A. Watt, *Fundamentals of 3D-Computergraphics*

2. Geometrische Modellierung

2.1 Begriffs- und Modellbildung

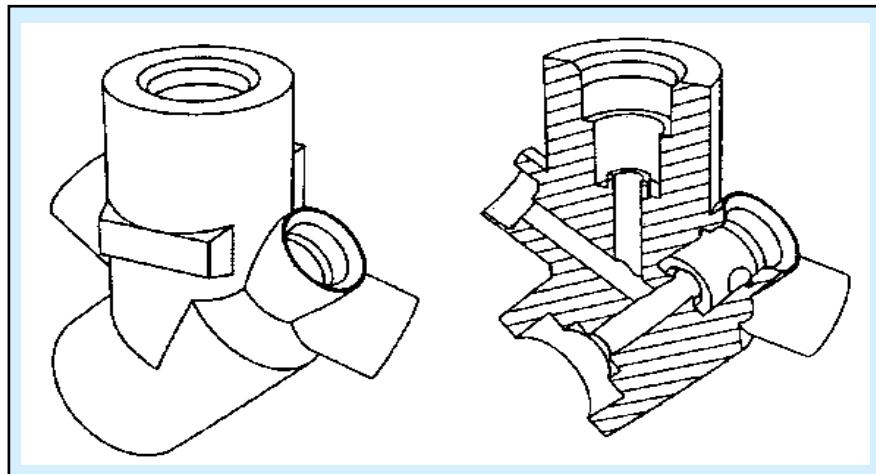


■ **Körpermodelle**

- beschreiben die Objekte als komplette, körperhafte Struktur.
- Für jeden 3D-Punkt kann entschieden werden, ob er sich innerhalb oder ausserhalb des Körpers befindet
- Bei Körpermodellen lassen sich Verdeckungs- und Schnittberechnungen durchführen.
- Sie sind die Basis zur Programmierung von NC-gesteuerten Werkzeugmaschinen für beliebig geformte Werkstücke.

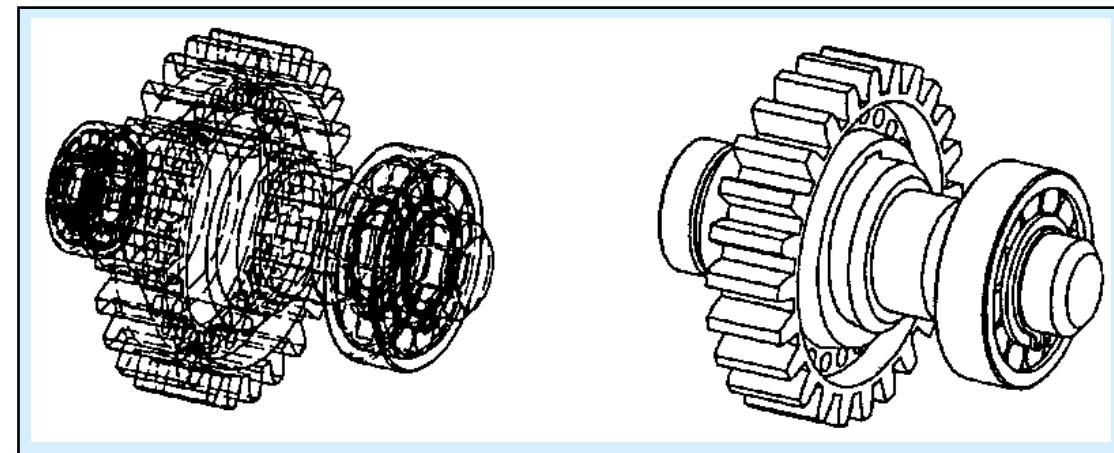
2. Geometrische Modellierung

2.1 Begriffs- und Modellbildung



*Beispiel für automatisches
Ausblenden verdeckter Kanten
beim Körpermodell*

*Ableitung eines Schnittes
aus einem Körpermodell
(Werkbild Strässle,
aus Ockert, 1993)*



2. Geometrische Modellierung

2.2 Körpermodellierung



- **Verschiedene Repräsentationen von Körpermodellen:**

- **Wir behandeln:**

- Boundary representation (B-Rep)
 - Constructive solid Geometry (CSG)
 - Octree

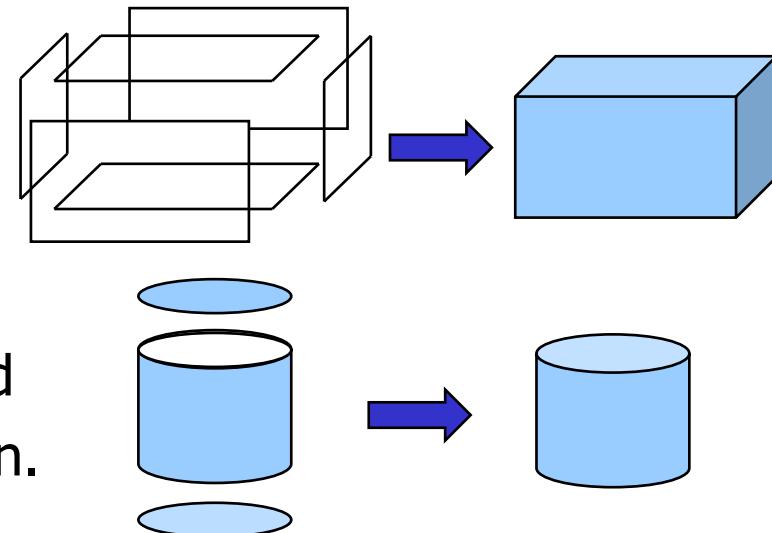


Boundary Repräsentation (B-Rep)

ist eine Graphenstruktur, die begrenzende Oberflächenteile eines Körpers und ihre Lage zueinander topologisch beschreibt.

Im Gegensatz zum Flächenmodell ist der von den Flächen umhüllte Raum materialerfüllt.

Im Gegensatz zu anderen Körpermodellen, sind Flächen, Kanten und Punkte explizit im Modell vorhanden.





- **Die B-Rep baut Körper aus folgenden Elementen auf:**

- **Schale (shell)**

ist eine aus Einzelteilen (Facetten) zusammengesetzte, zusammenhängende geschlossene Oberfläche.

- **Fläche (Facette)**

ist begrenzt von einer oder mehreren Konturen.

- **Kontur (loop)**

ist eine abgeschlossene, orientierte Folge von Kanten.

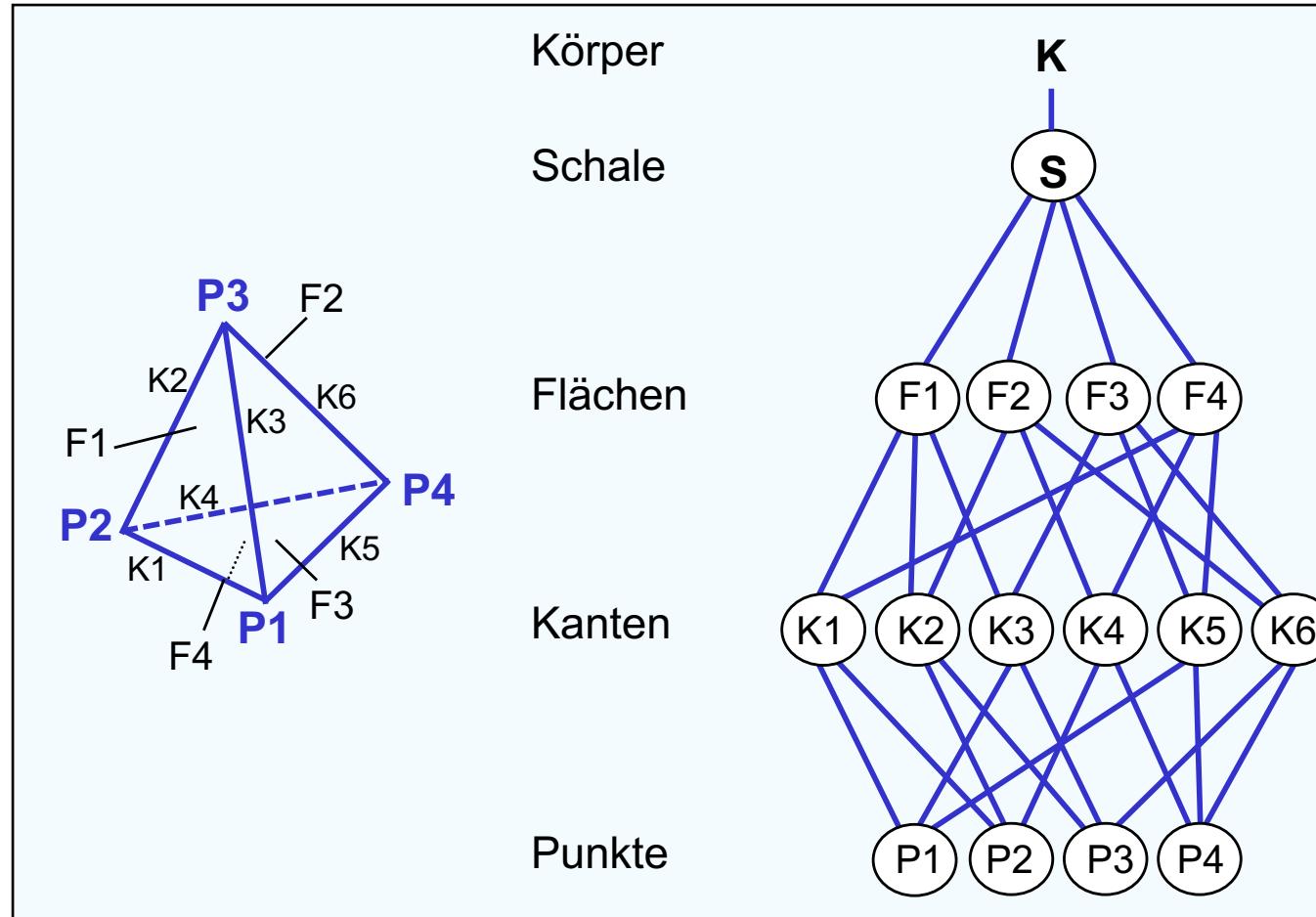
- **Kante (edge)**

ist ein orientiertes Geraden- (Kurven-) Stück, begrenzt durch zwei

- **Knoten (vertices).**

2. Geometrische Modellierung

2.2 Körpermodellierung



- *Boundary-Repräsentation für ein Tetraeder*

2. Geometrische Modellierung

2.2 Körpermodellierung



Bedingungen zur topologischen Gültigkeit von B-Reps:

- notwendige Bedingung:
Die Euler-Poincaré-Formeln müssen erfüllt sein.

Euler: $V - E + F = 2$

Euler Poincaré: $V - E + F - (L - F) - 2(S - G) = 0$

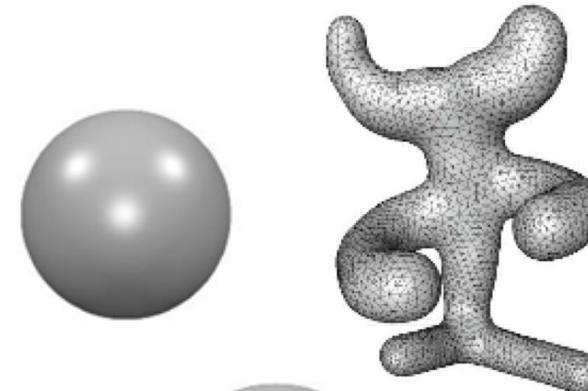
(V : Anzahl der Knoten, E : Anzahl der Kanten, F : Anzahl der Flächen, G : Geschlecht,
 L : Anzahl der Konturen, S : Anzahl der Schalen)

- hinreichende Bedingungen:
 - Die Oberfläche eines Körpers muss geschlossen sein.
 - Die Oberfläche eines Körpers muss orientierbar sein.
 - Die Oberfläche eines Körpers darf sich nicht schneiden.



- Genus (Geschlecht): Anzahl der Löcher in einem Körper

- Genus 0: Kugel, Würfel,...



- Genus 1: Torus, Kaffeetasse...



- Genus 2: Utah teapot (ohne Deckel), Topf mit 2 Henkeln, Doppeltorus...



2. Geometrische Modellierung

2.2 Körpermodellierung



Beispiele zur Demonstration der Formeln:

$V-E+F = 8-12+6 = 2$ $V-E+F = 8-13+7 = 2$

2 Objekte, die die Eulerformel erfüllen

Körper mit Bohrung und Loch erfüllt die allgemeine Euler-Poincaré-Formel

$$V - E + F - (L-F) - 2(S-G) = 0 = 24 - 36 + 16 - (18-16) - 2(2-1)$$



■ **CSG-Repräsentationen (Constructive Solid Geometry)**

erzeugen komplexe Körper durch Verknüpfen von Grundkörpern über Boolesche Mengenoperationen.

Grundkörper sind regelmäßige geometrische Primitivkörper (wie Quader, Kugel, Kegel, Zylinder, Torus, Pyramide u.ä.),

Boolesche Mengenoperationen sind:

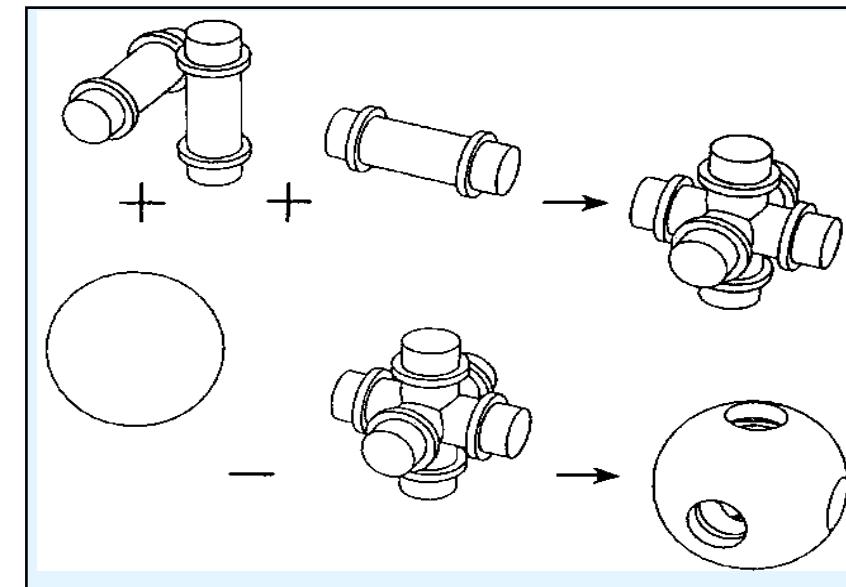
Vereinigung \cup

Durchschnitt \cap

Differenz \setminus



- Beschreibung von CSG-Modellen:
 - CSG-modellierte Körper werden durch Bäume, so bezeichnete CSG-Bäume, beschrieben.
 - Alle Knoten des CSG-Baumes, die mindestens einen Nachfolger haben, repräsentieren regularisierte Boolesche Operationen bzw. Transformationen im Raum.
 - Die Blätter des Baumes verweisen auf Grundkörper.
- Die Konstruktion eines CSG-Baumes ist eng mit der Konstruktion des Objektes verknüpft.

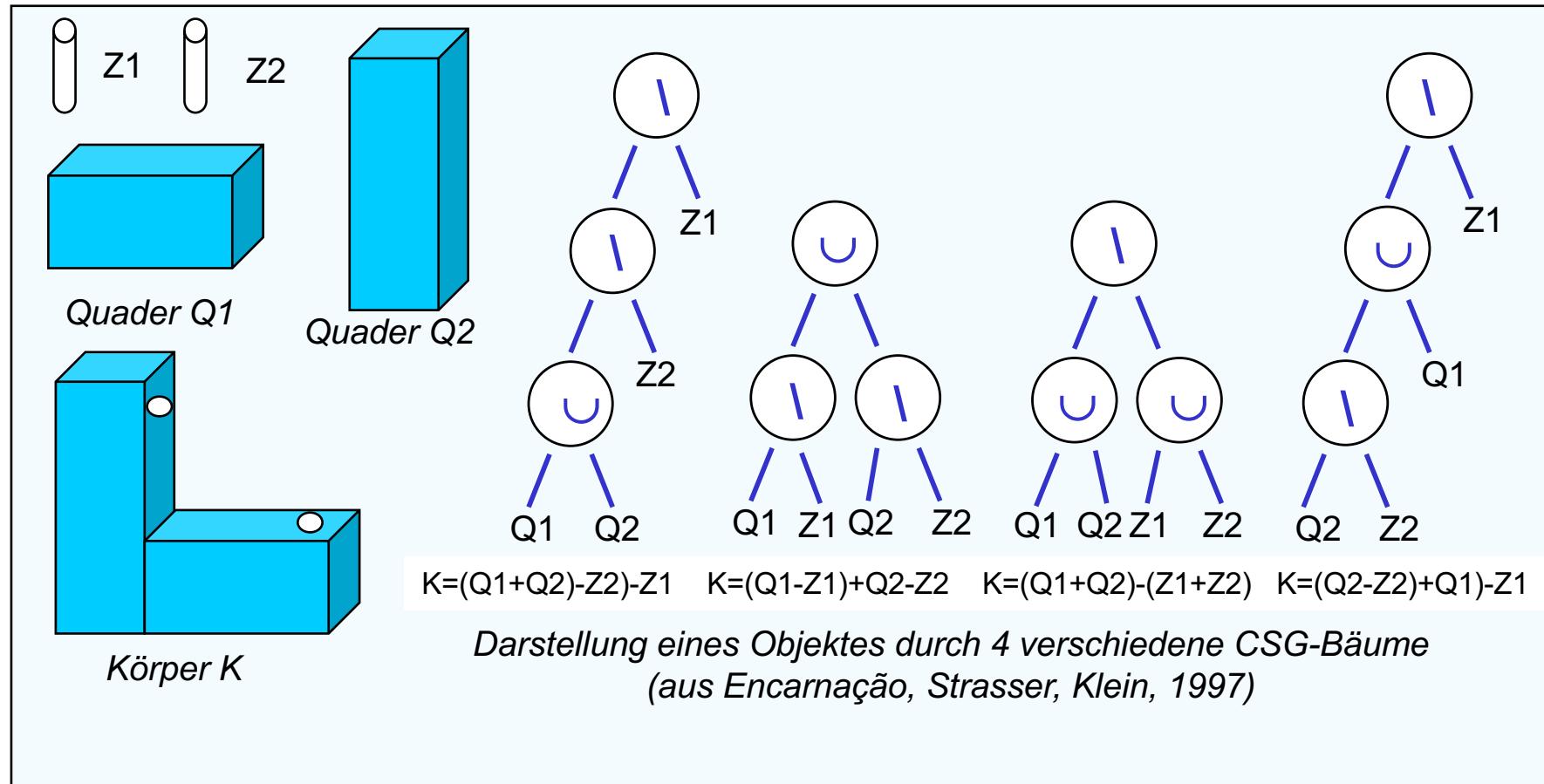


2. Geometrische Modellierung

2.2 Körpermodellierung

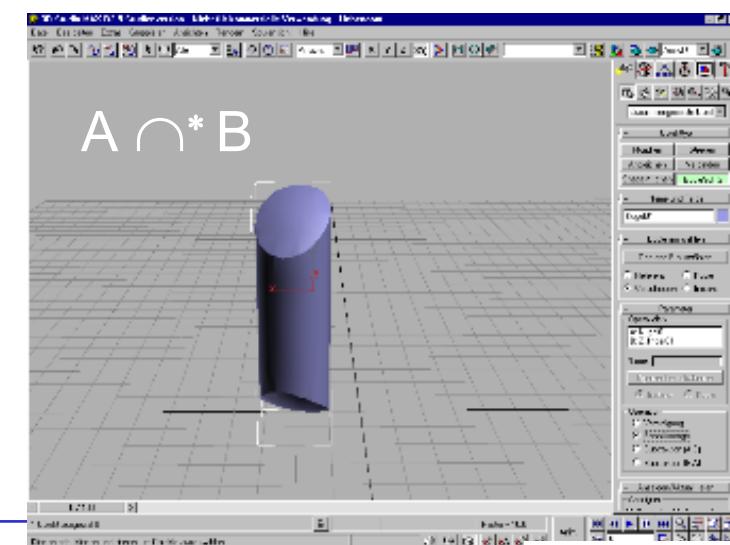
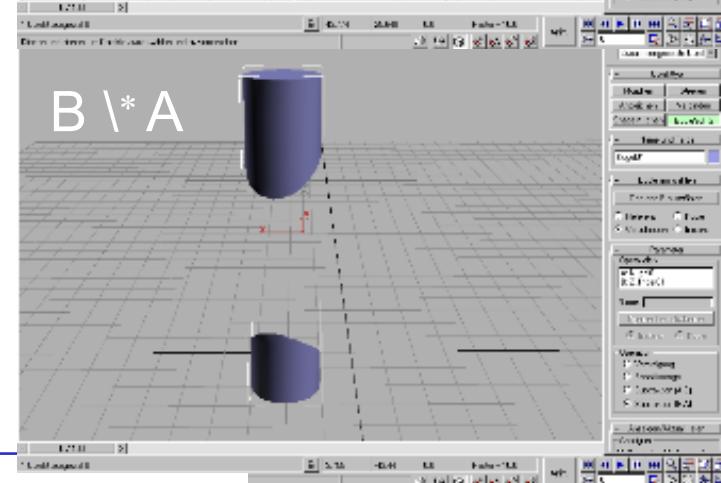
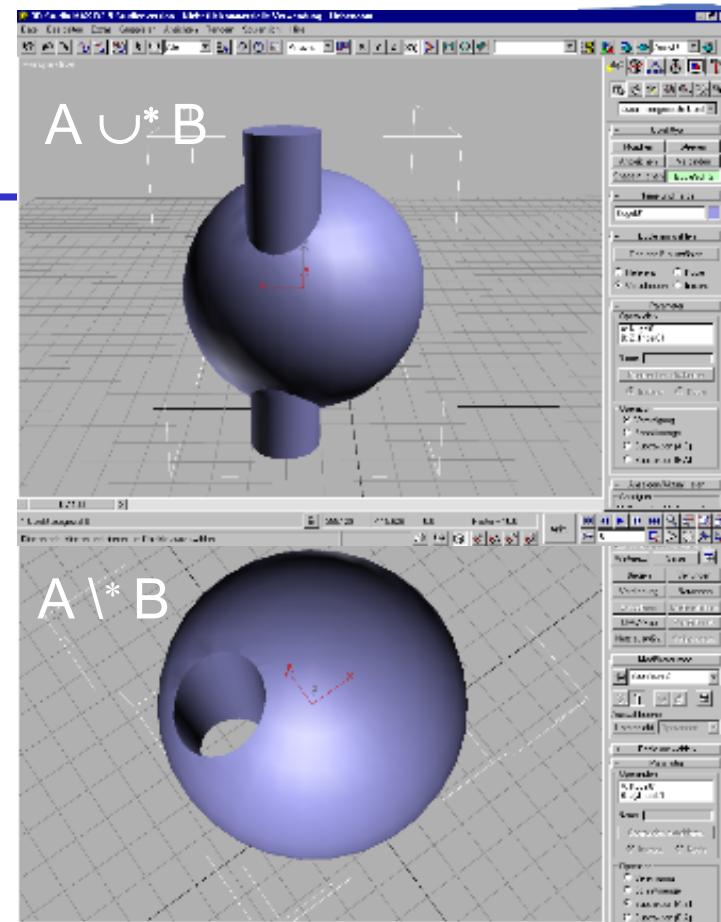
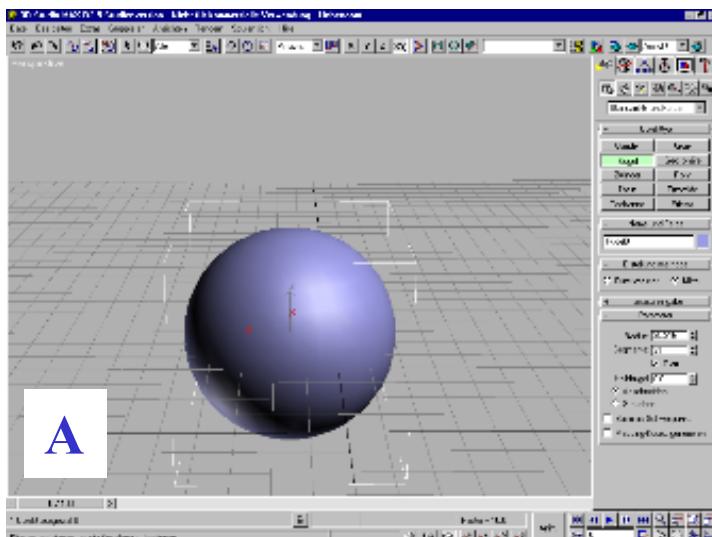


Die Konstruktion eines CSG-Baumes ist nicht eindeutig.



2. Geometrische Modellierung

2.2 Körpermodellierung



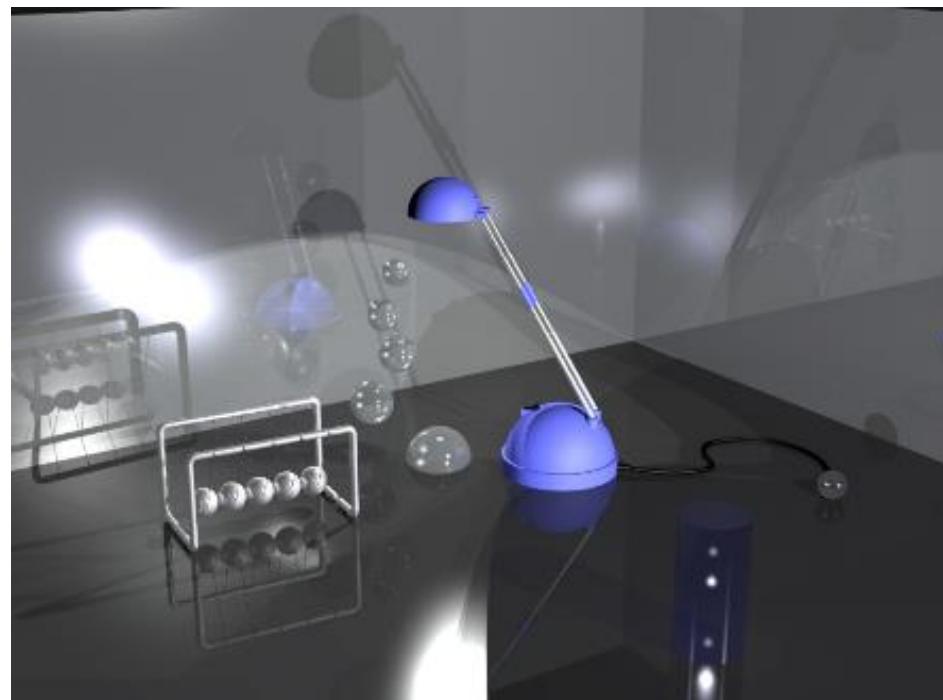
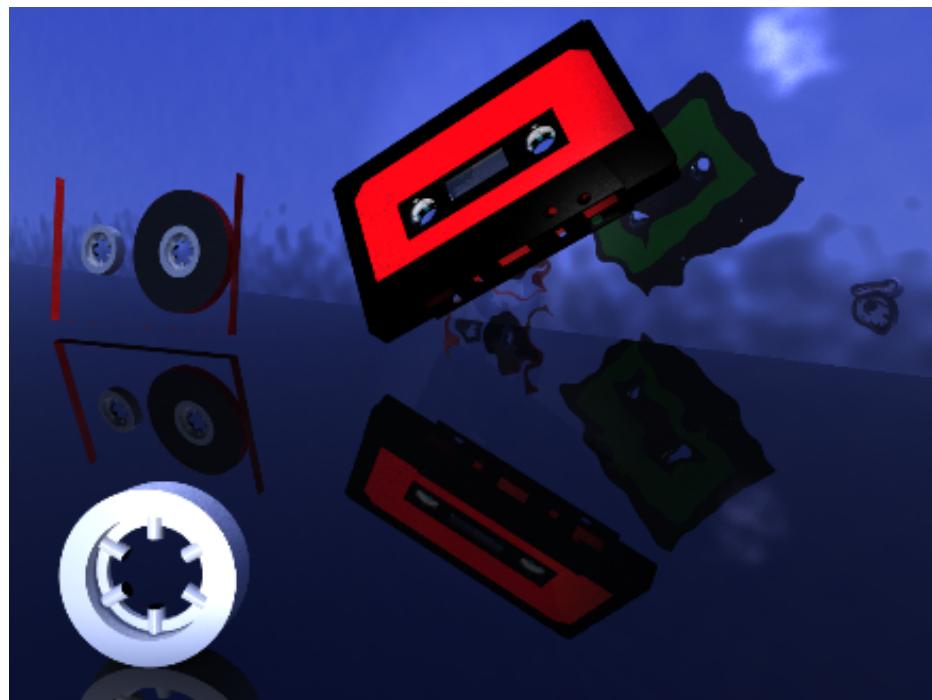
CSG-Modellierung mit 3D-Studio

2. Geometrische Modellierung

2.2 Körpermodellierung



CSG-modellierte Szenen





- **Octrees**

Erklärung in 2D (Quadtree):



■ Octrees

werden durch rekursive Unterteilung eines Würfels im E^3 in 8 Oktanten erzeugt. (Im E^2 werden analog durch rekursive Unterteilung eines Quadrates in 4 Quadranten Quadtrees gebildet.)

■ Die Wurzel eines Octrees ist der umschreibende Quader eines Körpers.

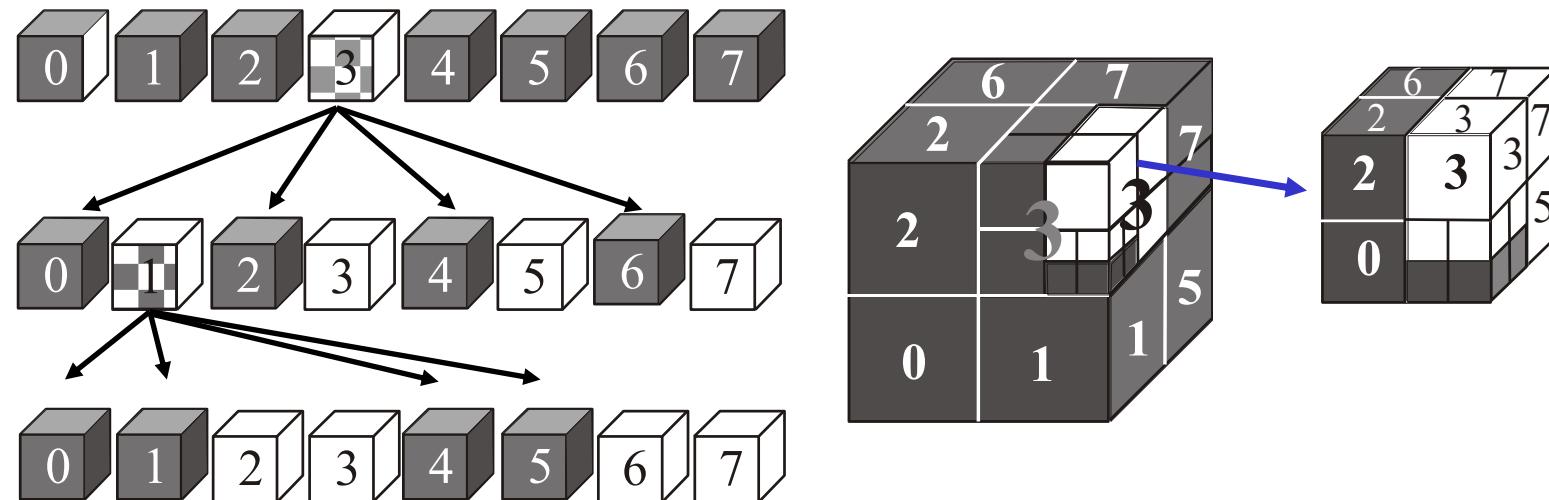
Jeder Knoten des Baumes wird markiert als:

- **voll** (schwarz): Der durch den Knoten repräsentierte Quader ist vollständig im Körper enthalten.
- **leer** (weiß): Der durch den Knoten repräsentierte Quader liegt vollständig außerhalb des Körpers.
- **partiell** (grau): Der durch den Knoten repräsentierte Quader liegt teilweise im Körper.



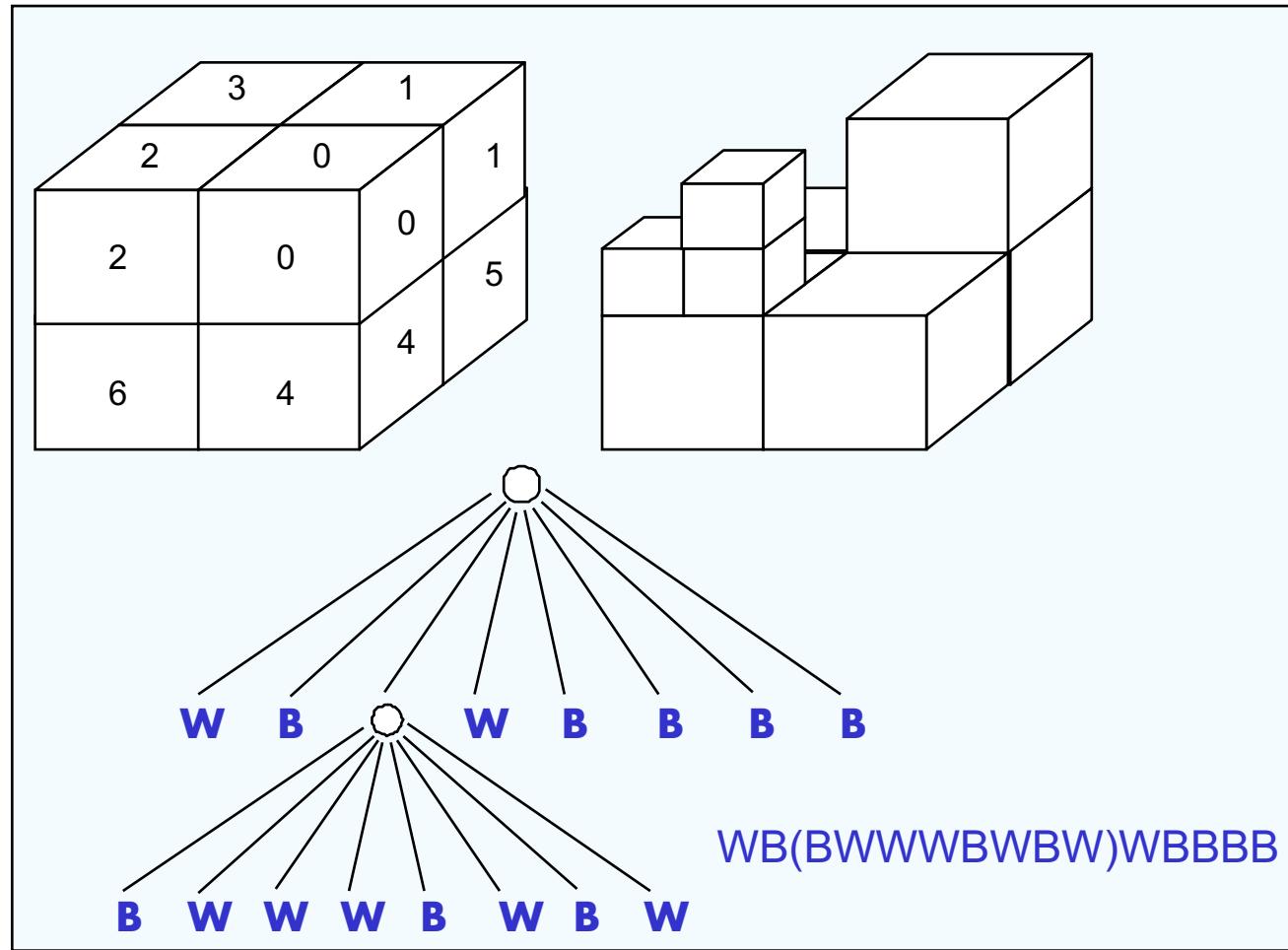
■ Aufbau des Octree

- Die Quader mit der Markierung "partiell" werden in Oktanten unterteilt und es werden Bögen vom unterteilten Quader zu jedem Oktanten gezogen.
- Die Oktanten werden markiert und gegebenenfalls weiter unterteilt.
- Der Prozeß wird so lange fortgesetzt, bis alle Oktanten entweder mit "weiß" oder "schwarz" markiert sind oder eine gegebene Rekursionstiefe erreicht ist.



2. Geometrische Modellierung

2.2 Körpermodellierung



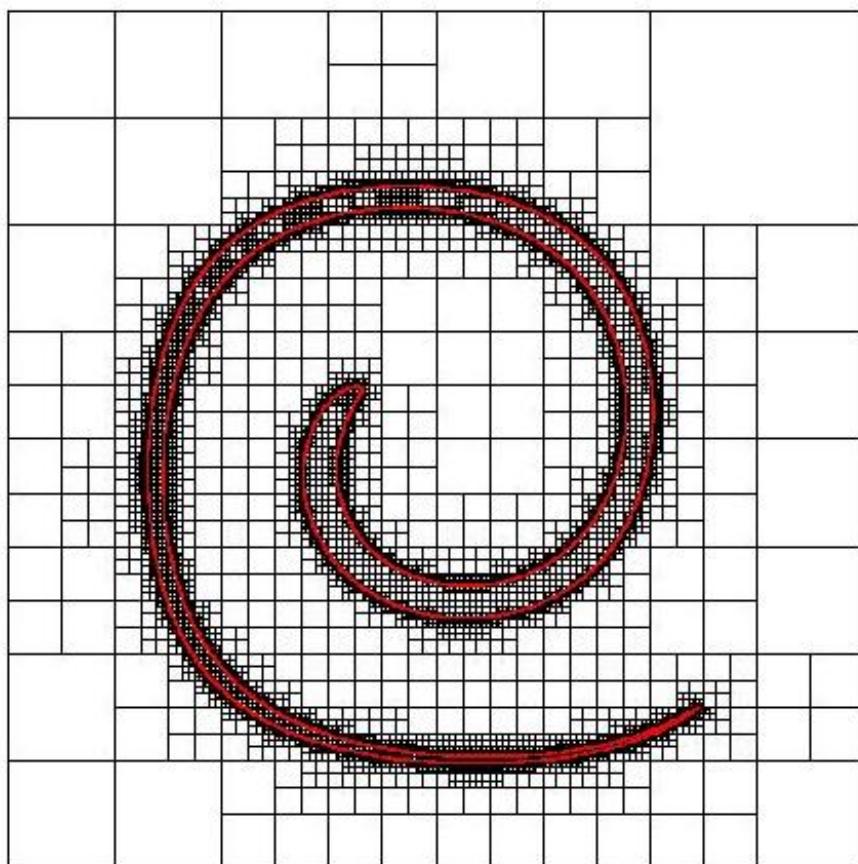
Beispiel zur Konstruktion und linearen Speicherung eines Octree

2. Geometrische Modellierung

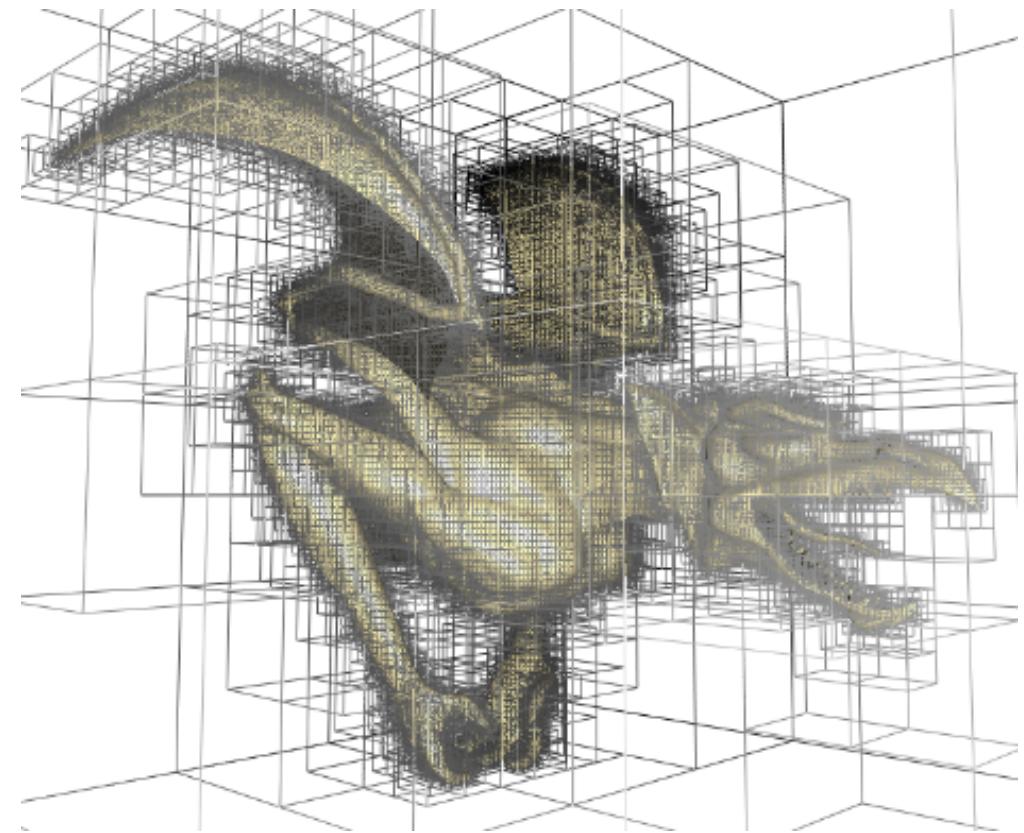
2.2 Körpermodellierung



Beispiel fuer Quadtree/Octree repräsentation



<http://www-evasion.imag.fr/Membres/Sylvain.Lefebvre/pict1.png>



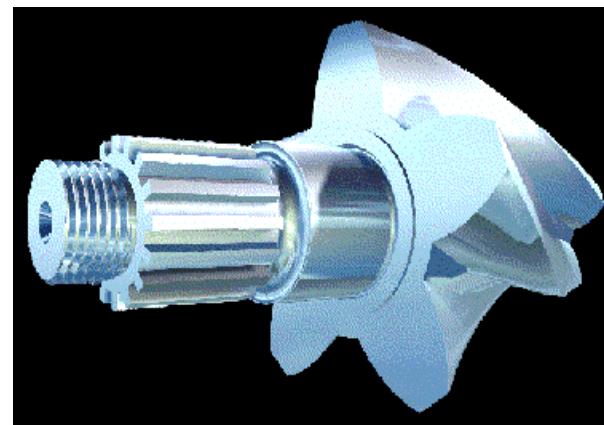
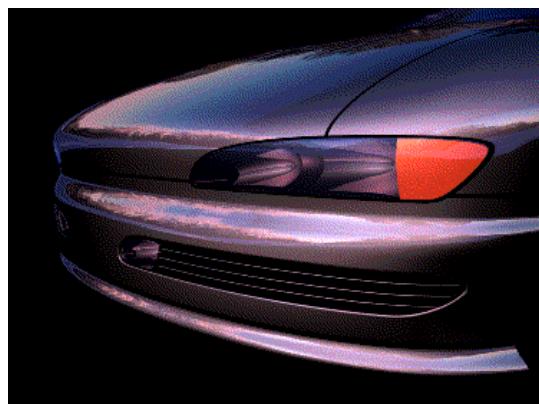
http://www.math.ucsb.edu/~chohong/vortex_in_a_box.jpg

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



- Viele Anwendungen benötigen die Möglichkeit zur Darstellung beliebiger Kurven und Flächen
- Typische Beispiele sind:
 - Autokarosserieentwurf, Flugzeug- und Schiffbau,
 - Simulation von Bewegungsabläufen,
 - Animation, Filmindustrie und Computerkunst.



Modellierung von Objekten mit Freiformflächen

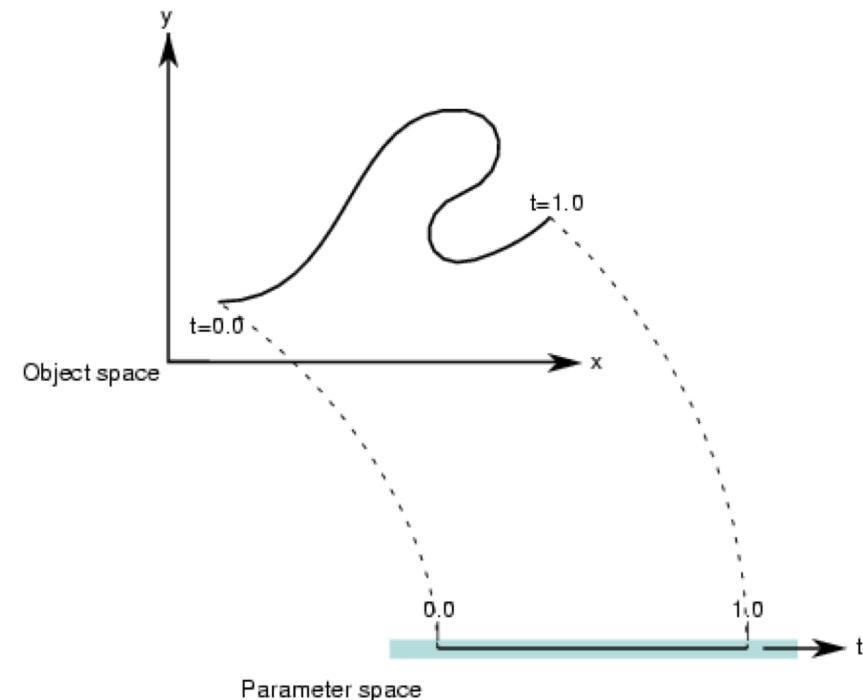


- **Freiformkurven/ Freiformflächen:**
- Parametrische Kurven/Flächen mit intuitiven Kontrollpunkten

$$\mathbf{x} : D \rightarrow \mathbb{R}^3$$

$$t \rightarrow \mathbf{x}(t) = \begin{pmatrix} x(t) \\ y(t) \\ z(t) \end{pmatrix}.$$

Parametrische Kurve

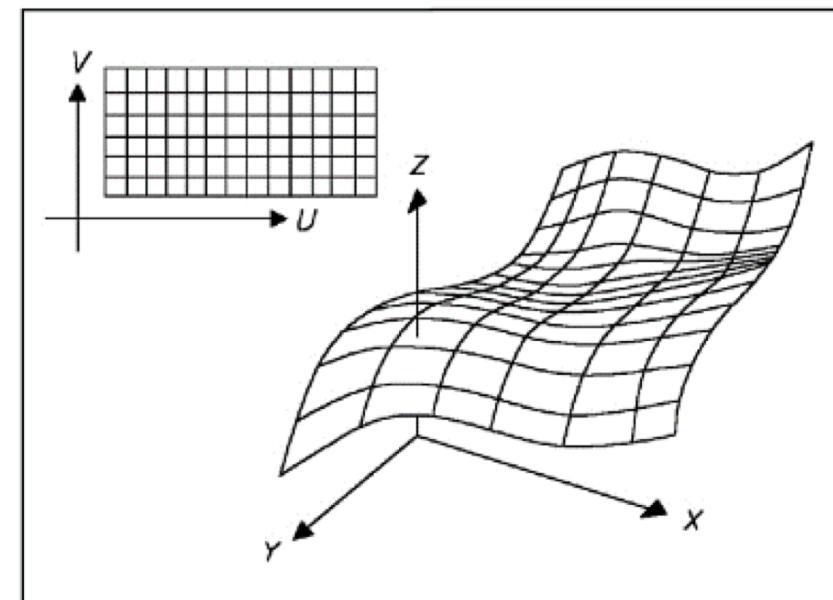




- **Freiformkurven/ Freiformflächen:**
- Parametrische Kurven/Flächen mit intuitiven Kontrollpunkten

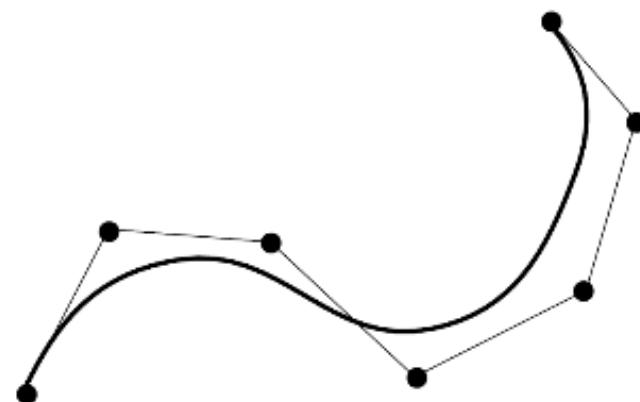
$$\mathbf{x} : D \rightarrow \mathbb{R}^3 \\ (u, v) \rightarrow \mathbf{x}(u, v) = \begin{pmatrix} x(u, v) \\ y(u, v) \\ z(u, v) \end{pmatrix}.$$

Parametrische Fläche

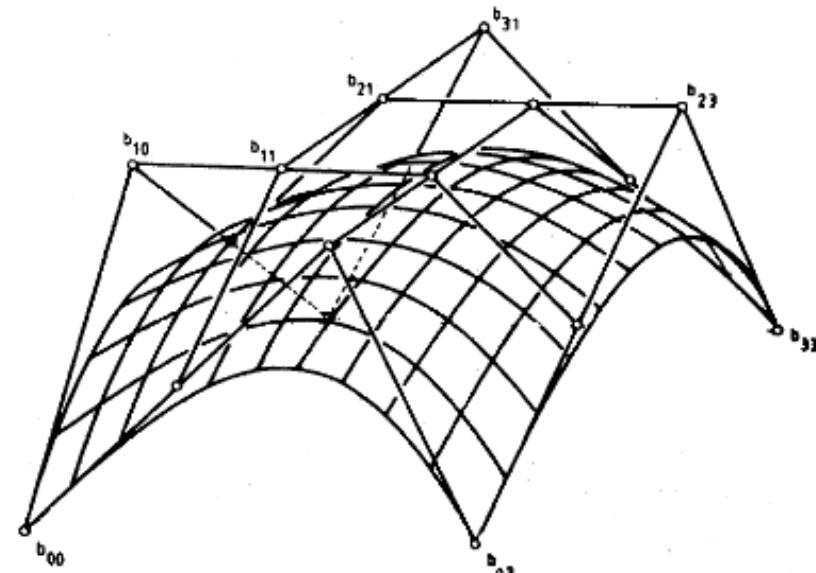




- **Freiformkurven/ Freiformflächen:**
- Parametrische Kurven/Flächen mit intuitiven Kontrollpunkten



Freiformkurve



Freiformfläche

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



- Freiformkurven/flächen können auf zwei Arten definiert werden

- **algebraisch**

- Durch Definition von $n-1$ Basisfunktion $f_i(t)$, $i=0,..,n$:
 - dann wird durch die Kontrollpunkte \mathbf{p}_i die Kurve definiert als

$$\mathbf{x}(t) = \sum_{i=0}^n \mathbf{p}_i f_i(t)$$

- **geometrisch**

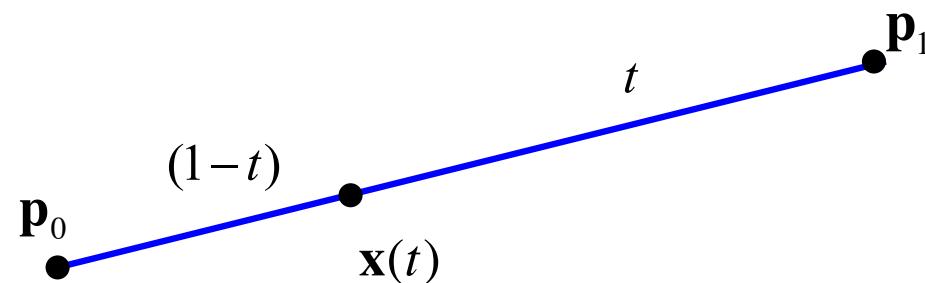
- Beschreibung einer geometrischen Konstruktion, wie aus den Kontrollpunkten \mathbf{p}_i und dem Parameterwert t der entsprechende Kurvenpunkt konstruiert werden kann



■ **Der de Casteljau-Algorithmus**

- Geometrisches Verfahren zur Konstruktion von Freiformkurven
- Basiert auf wiederholter linearer Interpolation

$$\mathbf{x}(t) = (1-t)\mathbf{p}_0 + t \cdot \mathbf{p}_1, \quad t \in [0, 1]$$



Lineare Interpolation



■ Der de Casteljau-Algorithmus

geg.: $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n \in E^3$ und $t \in \mathbb{R}$

setze: $\mathbf{b}_i^r = (1 - t) * \mathbf{b}_i^{r-1}(t) + t * \mathbf{b}_{i+1}^{r-1}(t)$

$r = 1, \dots, n, \quad i = 0, \dots, n-r \quad \text{und} \quad \mathbf{b}_i^0(t) = \mathbf{b}_i$

Dann ist $\mathbf{b}_0^n(t)$ der gesuchte Kurvenpunkt, der dem Parameterwert t entspricht.

Die konstruierte Kurve heisst Bezierkurve

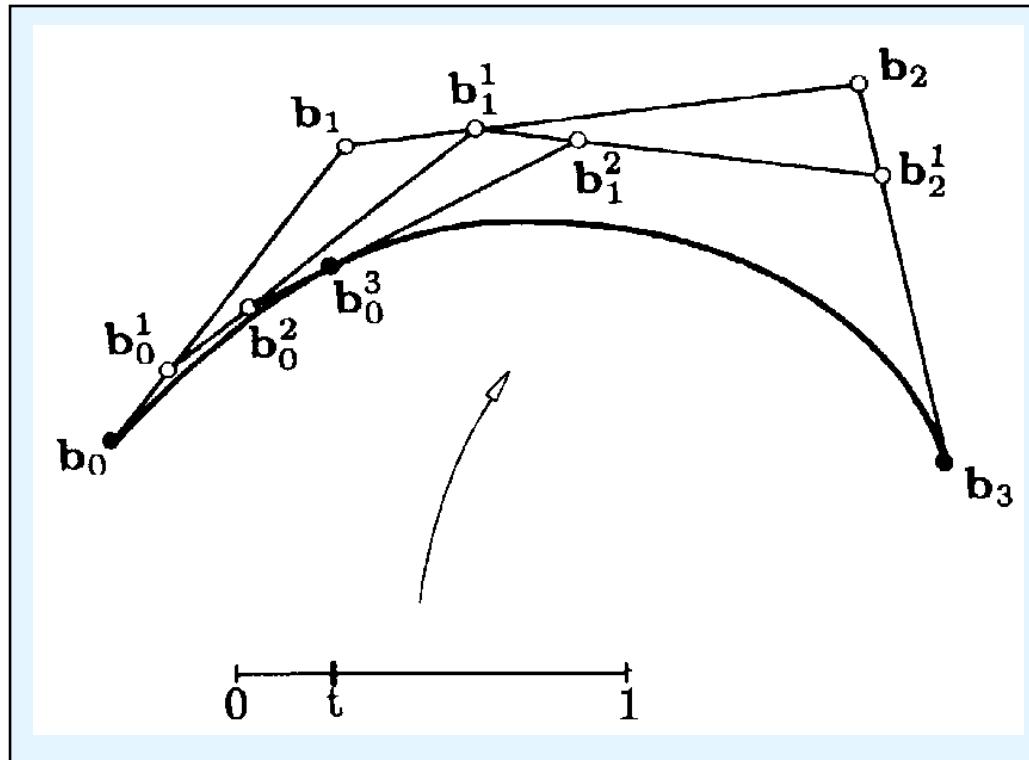
Die Punkte \mathbf{b}_i heissen Bezierpunkte



- **Der de Casteljau-Algorithmus**

2. Geometrische Modellierung

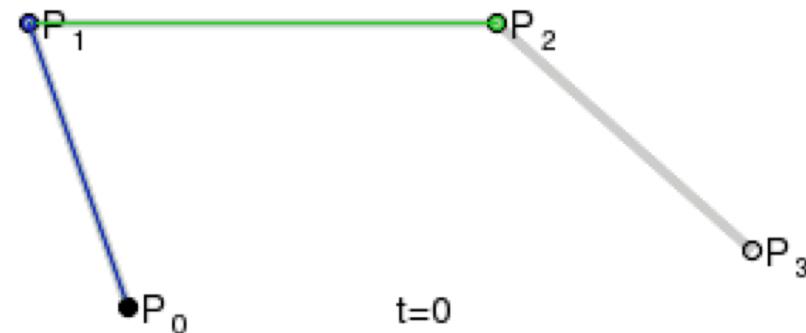
2.3 Kurven- und Flächenmodellierung



Anwendung des de Casteljau-Algorithmus
zur Konstruktion einer kubischen Bezier-Kurve
(b_3^3 wird durch wiederholte lineare Interpolation für $t = \frac{1}{4}$ berechnet)

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung





■ Die Bernsteinform einer Bezierkurve

Bezierkurven können auch algebraisch beschrieben werden

$$\mathbf{x}(t) = \sum_{i=0}^n \mathbf{b}_i B_i^n(t)$$

mit

- t Parameter ($0 \leq t \leq 1$)
- n Grad der Bezierkurve
- \mathbf{b}_i **Kontrollpunkte** aus \mathbb{R}^2
oder \mathbb{R}^3 mit ($0 \leq i \leq n$)
- $B_i^n(t)$ Bernsteinpolynome
vom Grad n über dem
Intervall $[0, 1]$

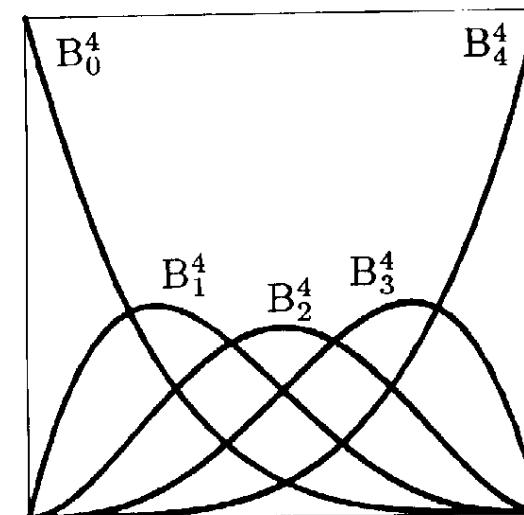


- Bernsteinpolynome sind explizit definiert durch:

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

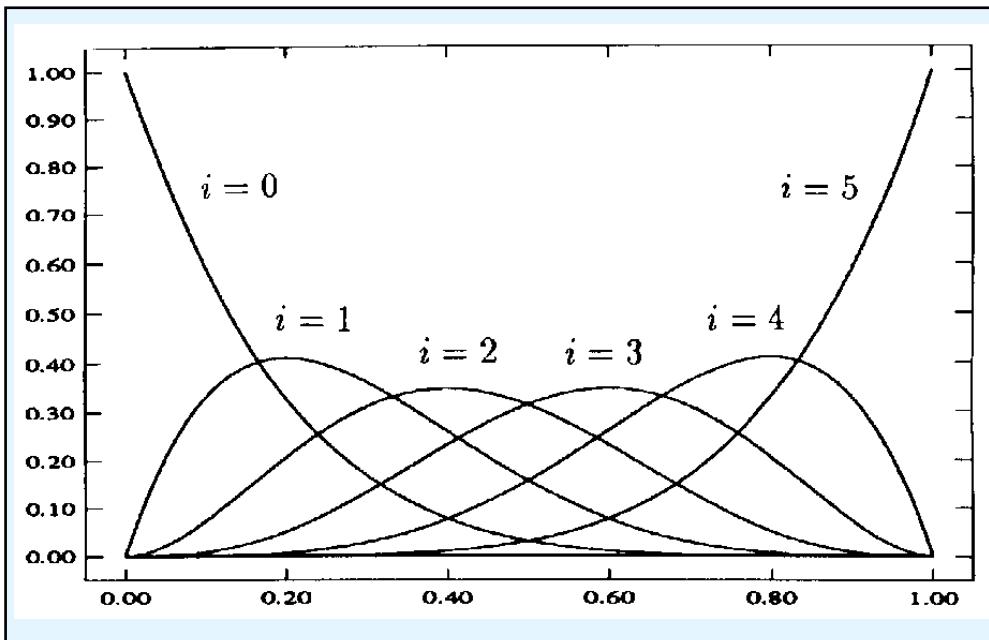
wobei die Binominialkoeffizienten gegeben sind durch:

$$\binom{n}{i} = \begin{cases} \frac{n!}{(n-i)!i!} & \text{falls } 0 \leq i \leq n \\ 0 & \text{sonst} \end{cases}$$

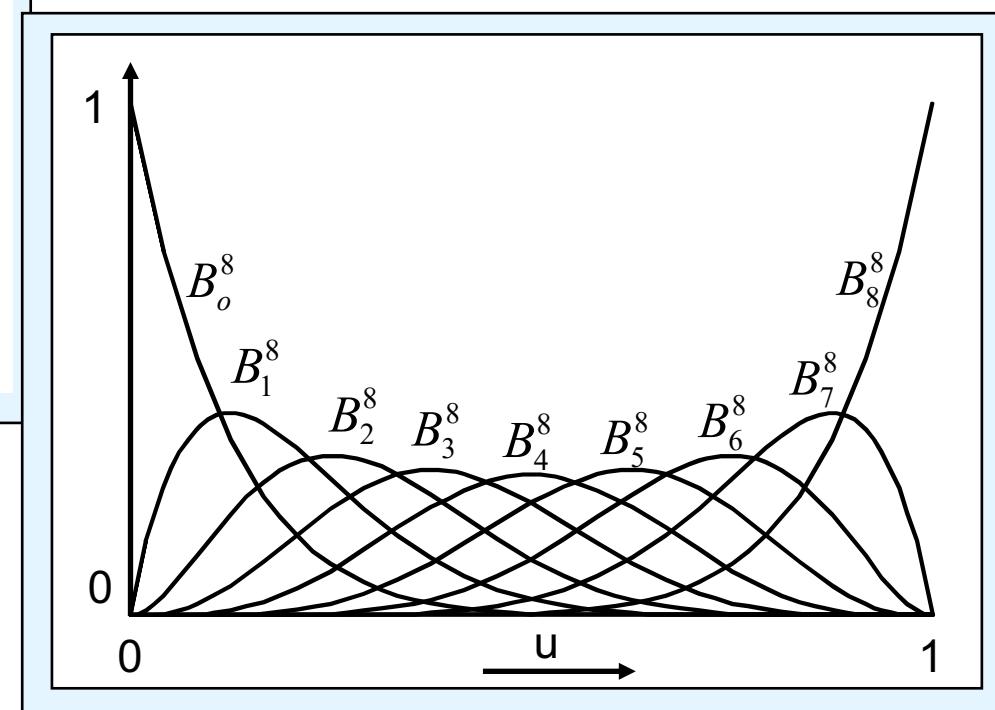


2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



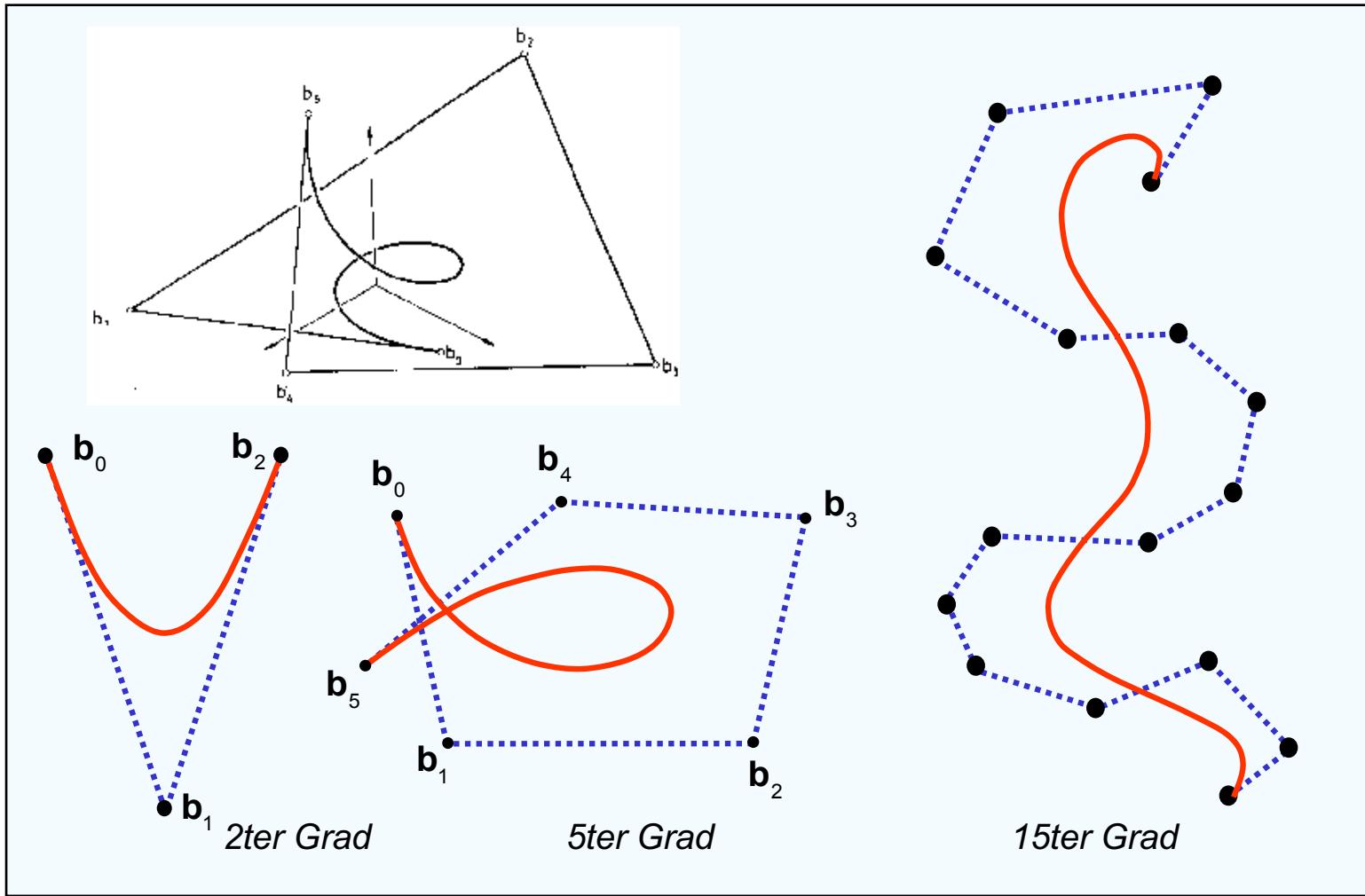
Bernsteinpolynome 5. Grades



Bernsteinpolynome 8. Grades

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



Verschiedene Bezierkurven mit zugehörigen Kontrollpolygone

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



- Bezierkurven haben folgende Eigenschaften:
 - **Grad**
Der Grad der Bézier-Kurve ist abhängig von der Zahl der Polygonpunkte (*n* Punkte ergeben den Grad *n*-1).
 - **Globale Wirksamkeit**
Die Modifikation eines Polygonpunktes wirkt sich global auf die Bézier-Kurve aus.
 - **Einfluß von Kontrollpunkten**
 \mathbf{b}_i hat den größten Einfluß auf $\mathbf{x}(t)$ bei $t = i/n$.
 - **Symmetrie**
Es ist unerheblich, ob wir die Bezierpunkte mit $\mathbf{b}_0, \mathbf{b}_1, \dots, \mathbf{b}_n$ oder mit $\mathbf{b}_n, \mathbf{b}_{n-1}, \dots, \mathbf{b}_0$ bezeichnen. Die Kurven stimmen überein, sie werden nur in unterschiedlicher Richtung durchlaufen.
 - **Affine Invarianz**
Die Bezierkurve ist unter affinen Abbildungen invariant.

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



- **Verhalten in den Eckpunkten**

Kontrollpolygon und Bezierkurve stimmen in Anfangs- und Endpunkt überein.

Das erste und letzte Segment des Kontrollpolygons verläuft in Anfangs- und Endpunkt tangential zur Bezierkurve.

Die inneren Bezierpunkte werden i.allg. nur approximiert.

- **Beschränkte Schwankung** (variation diminishing property)

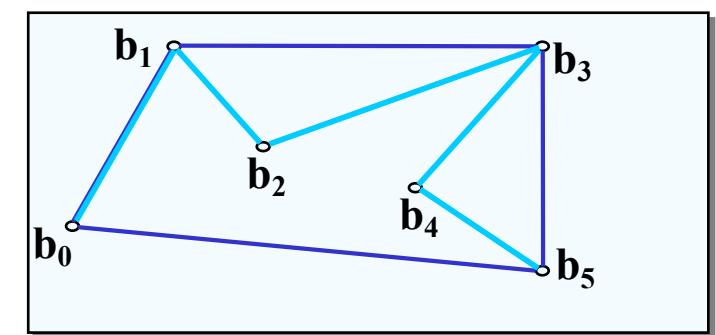
Keine Gerade schneidet die Bezierkurve öfter als das entsprechende Kontrollpolygon.

- **Graderhöhung**

Bei fortgesetzter Graderhöhung konvergiert das Kontrollpolygon gegen die Bezierkurve.

- **Konvexe-Hüllen-Eigenschaft**

Eine ebene Bezierkurve liegt ganz innerhalb der konvexen Hülle des zugehörigen Bezierpolygons.





■ Rationale Bezierkurven

- erlauben vielfältige weitere Gestaltungsmöglichkeiten, unter anderem auch die Darstellung von Kegelschnitten.
- Als wichtiges Designelement werden Gewichte w_i eingeführt, die den Einfluß eines Kontrollpunktes auf die Kurve steuern.
- Gegeben sind also die Kontrollpunkte \mathbf{b}_i und Gewichte w_i .
- Diese überführen wir in homogene Koordinaten mit:

$$\bar{\mathbf{b}}_i = \begin{pmatrix} w_i * \mathbf{b}_i \\ w_i \end{pmatrix} = \begin{pmatrix} w_i * x_i \\ w_i * y_i \\ w_i \end{pmatrix}$$

Wir berechnen nun die Kurve $\bar{\mathbf{x}}(t)$ in homogenen Koordinaten.

Die Zentralprojektion von $\bar{\mathbf{x}}(t)$ in die Ebene $z = 1$ ergibt die rationale Kurve $\mathbf{x}(t)$.

2. Geometrische Modellierung

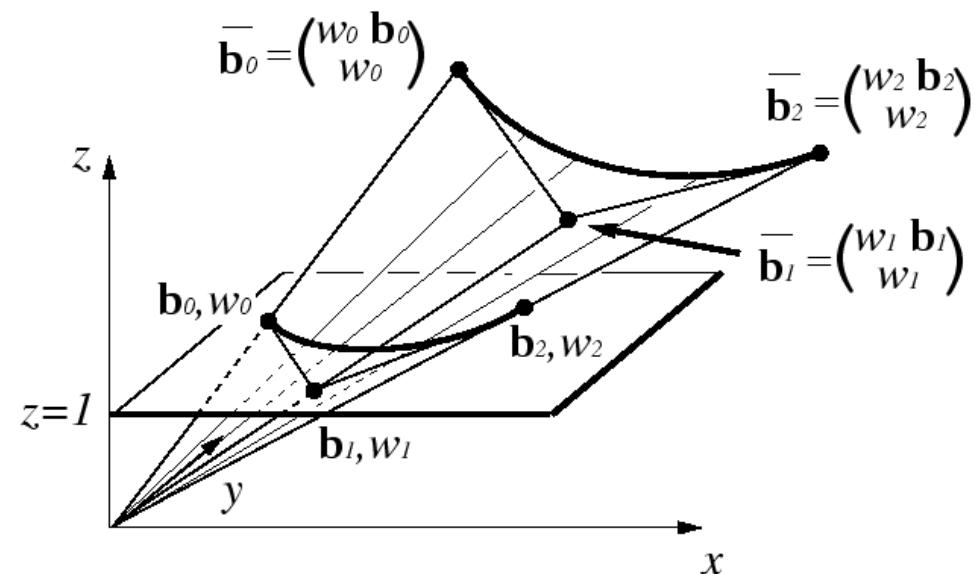
2.3 Kurven- und Flächenmodellierung



$\bar{\mathbf{x}}(t)$ in homogenen Koordinaten ist gegeben durch:
Durch Projektion erhalten wir $\mathbf{x}(t)$ mit:

$$\mathbf{x}(t) = \frac{\sum_{i=0}^n w_i \mathbf{b}_i B_i^n(t)}{\sum_{i=0}^n w_i B_i^n(t)}$$

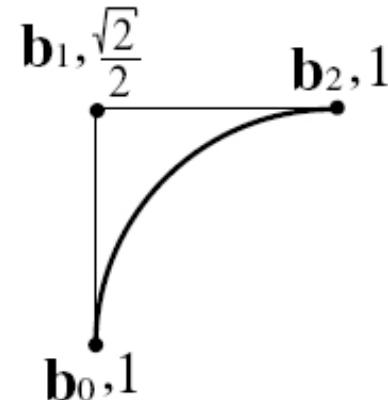
$$\bar{\mathbf{x}}(t) = \sum_{i=0}^n \bar{\mathbf{b}}_i * B_i^n(t) = \begin{pmatrix} \sum w_i * x_i * B_i^n(t) \\ \sum w_i * y_i * B_i^n(t) \\ \sum w_i * B_i^n(t) \end{pmatrix}$$





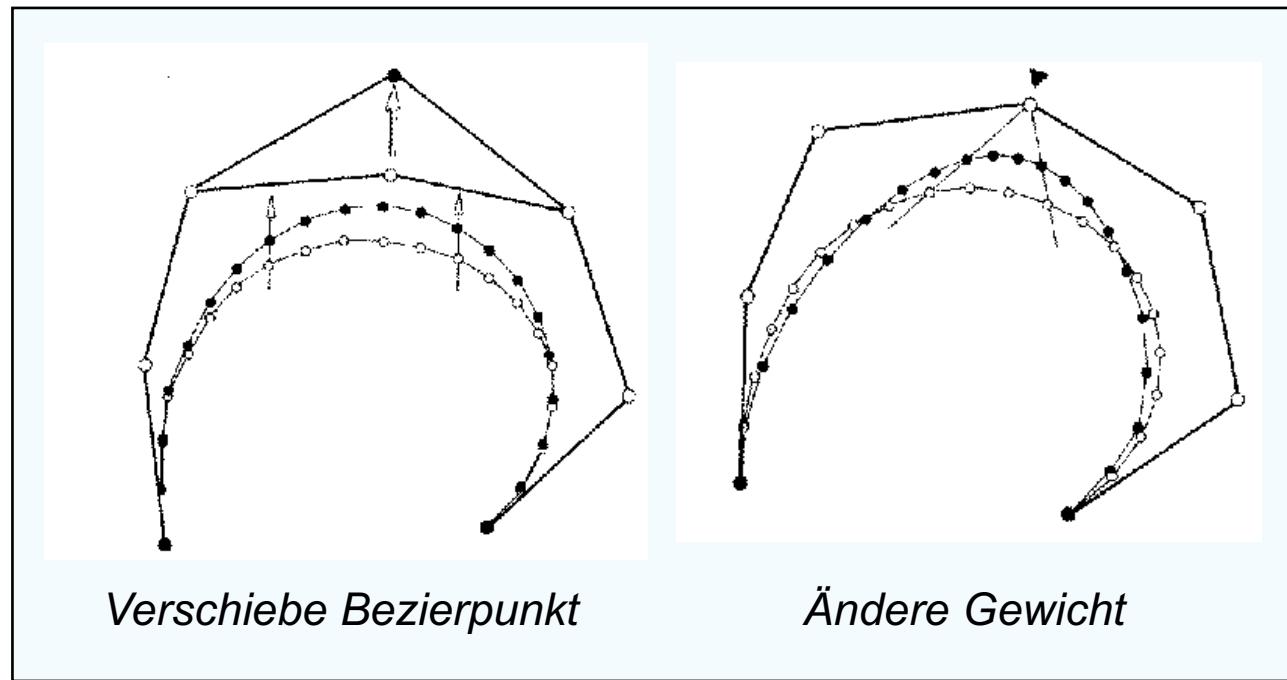
- **Beispiel:**

- $w_i = 1$ ($i = 0, \dots, n$): "normale" Bezierkurve
- Viertelkreis:



2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



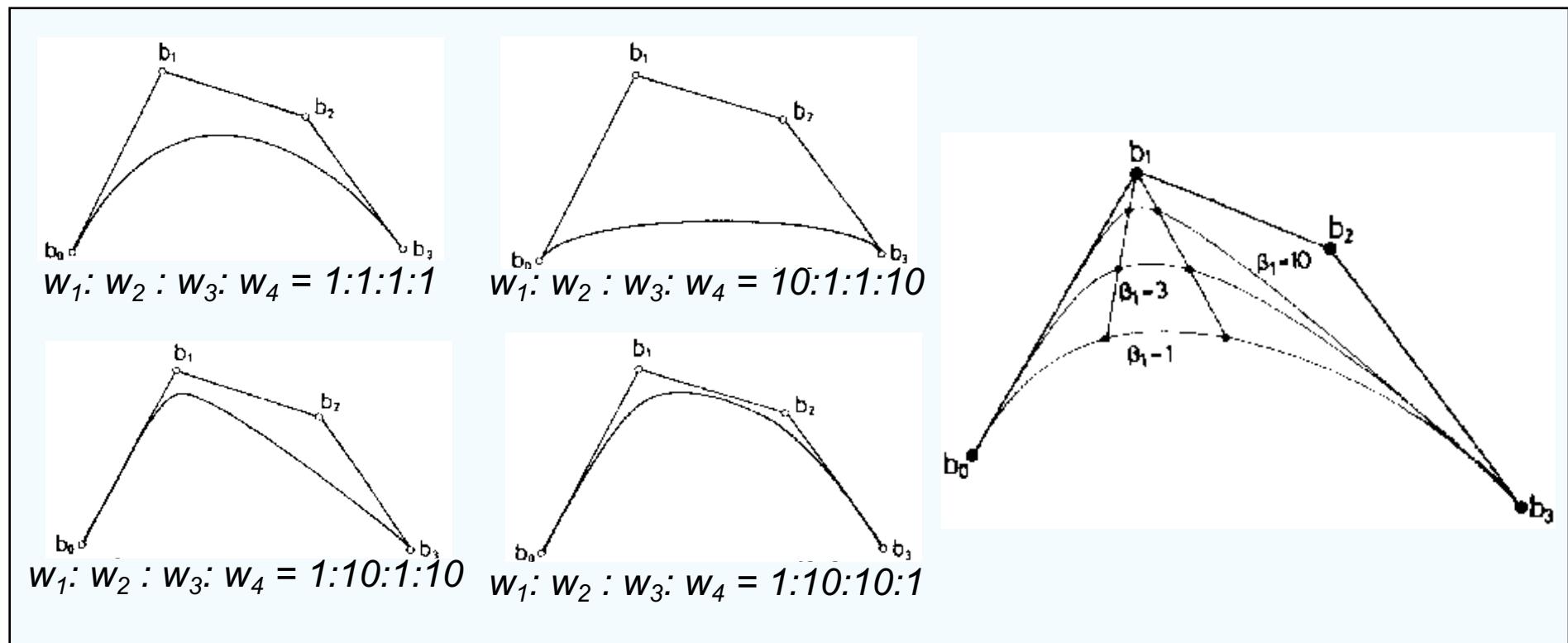
Einfluss der Gewichte auf den Kurvenverlauf

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



- Nicht die Gewichte entscheiden über den Kurvenverlauf, sondern die Gewichtsverhältnisse:



Veranschaulichung der Wirkung von Gewichten Gewichten

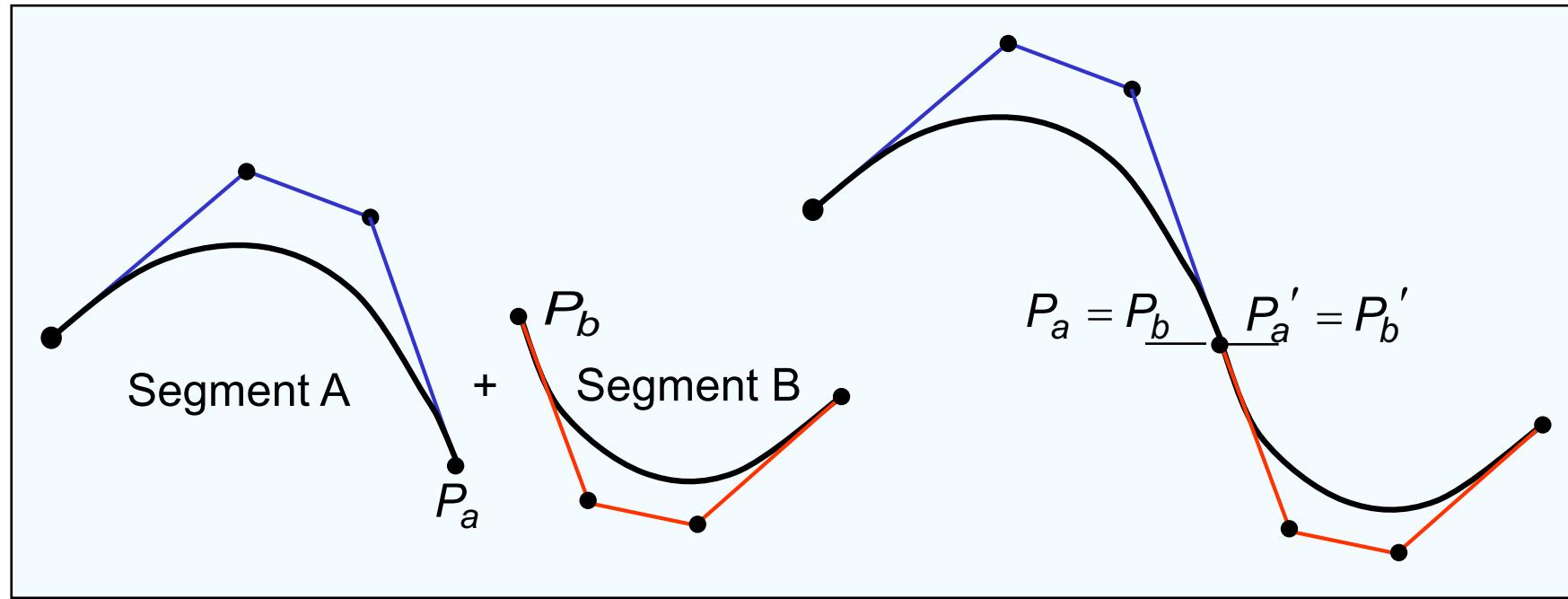


■ **Bezier-Spline-Kurve**

- Bei komplexen Formen wird eine größere Zahl von Bezierpunkten benötigt. Mit jedem Bezierpunkt erhöht sich der Grad um 1.
- Um den Grad niedrig zu halten und trotzdem keine Begrenzung bezüglich der Kontrollpunkte zu haben, ist es notwendig, einzelne Bezier-Kurven-Elemente aneinander zu fügen.
- So erhalten wir eine Bezier-Spline-Kurve.
- Benachbarte Spline-Elemente müssen dieselbe Ordnung haben und gewisse Anschlußbedingungen einhalten, um einen stetigen Übergang zu garantieren.
- Bei einer kollinearen Ausrichtung der Anschlußkanten der Kontrollpolygone ist ein sog. G₁-stetiger (tangentenstetiger) Übergang garantiert

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung





■ Tensorprodukt-Bezierflächen

Die Definition von Bezierflächen erfolgt analog zur Definition der Kurven:

$$\mathbf{x}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{b}_{ij} B_i^n(u) B_j^m(v)$$

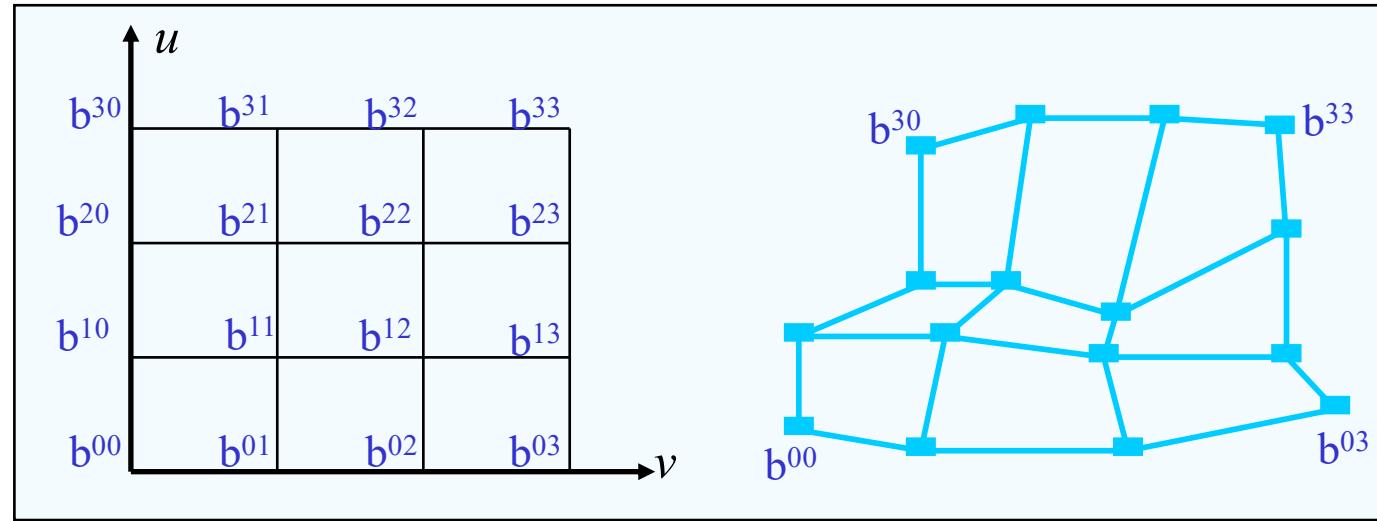
u und v bilden einen Parameterraum mit: $0 \leq u \leq 1, 0 \leq v \leq 1$,

\mathbf{b}_{ij} sind die Kontrollpunkte,

$B_r^s(t)$ sind die Bernstein-Polynome vom Grade s und
die Kontrollpunkte bilden in Zeilen und Spalten jeweils
Kontrollpolygone von Bezier-Kurven.

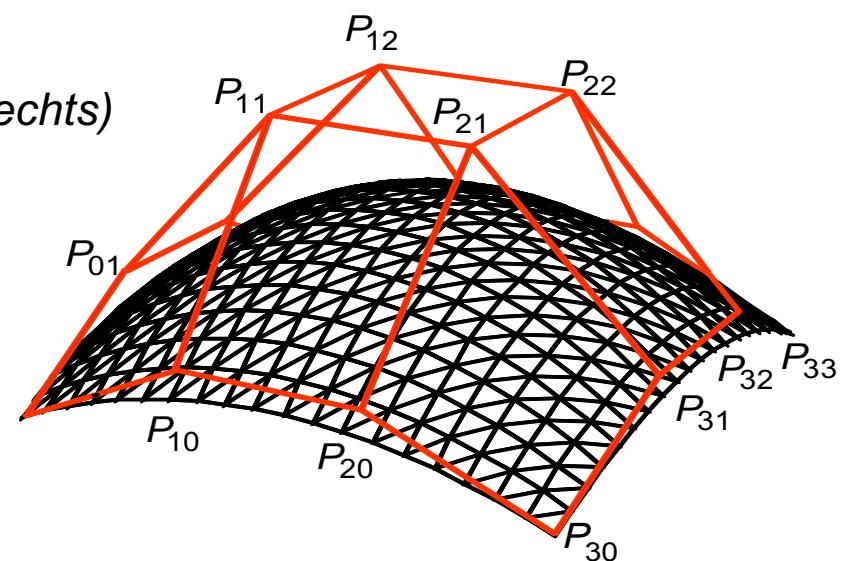
2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



*Kontrollnetz einer Bezierfläche ,
im Parameter-Raum (links) und im R^3 (rechts)*

*Kontrollnetz und Bezierfläche,
(unten)*





- Wichtige Eigenschaften der Bezierflächen sind:
 - Randpunkte
 b_{00} , b_{nm} , b_{0m} und b_{n0} interpolieren die Bezierfläche in diesen Punkten.
 - Konvexe Hülle
Die Bezierfläche liegt vollständig innerhalb der konvexen Hülle der Punkte des Kontrollnetzes.
 - Affine Invarianz
Kontrollnetz und Bezier-Fläche sind invariant gegenüber affinen Transformationen.
 - globale Wirksamkeit
Die Änderung eines Kontrollpunktes verändert die gesamte Fläche .
 - Geometrische Interpretierbarkeit
Punkte auf der Bezier-Fläche lassen sich mit dem de Casteljau-Algorithmus gewinnen.



■ De Casteljau für Bezierflächen

beruht auf wiederholter bilinearer Interpolation zwischen 4 Punkten

Zuerst Interpolation über u

$$\mathbf{p}' = (1-u) \cdot \mathbf{p}_{00} + u \cdot \mathbf{p}_{10}$$

$$\mathbf{p}'' = (1-u) \cdot \mathbf{p}_{01} + u \cdot \mathbf{p}_{11}$$

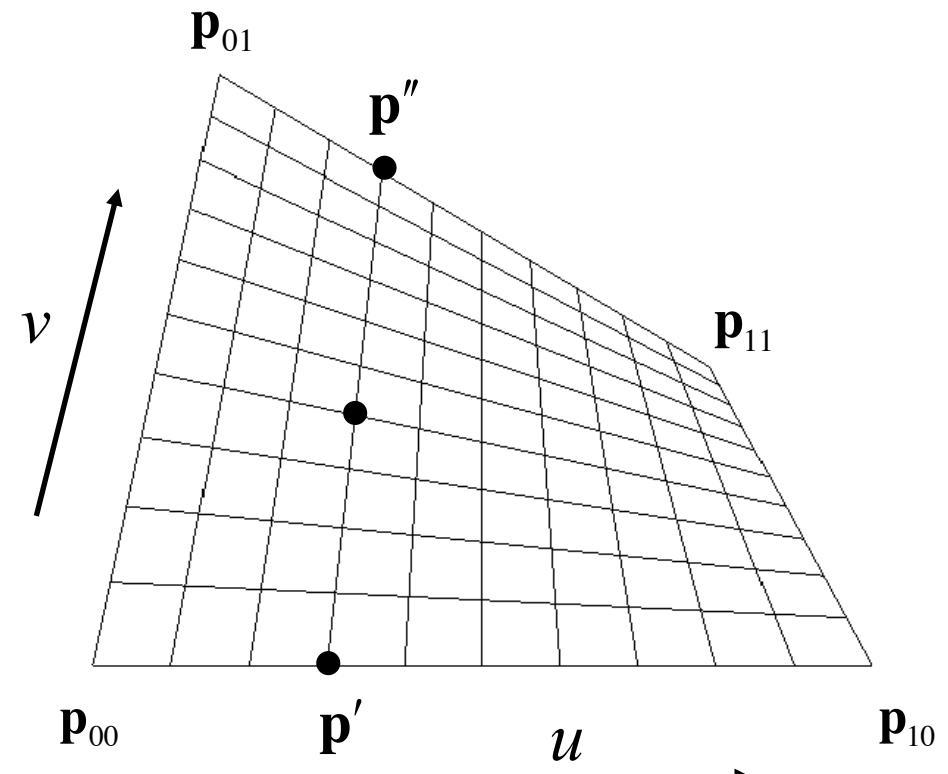
Dann über v

$$\mathbf{p}(u, v) = (1-v) \cdot \mathbf{p}' + v \cdot \mathbf{p}''$$

Zusammengefasst

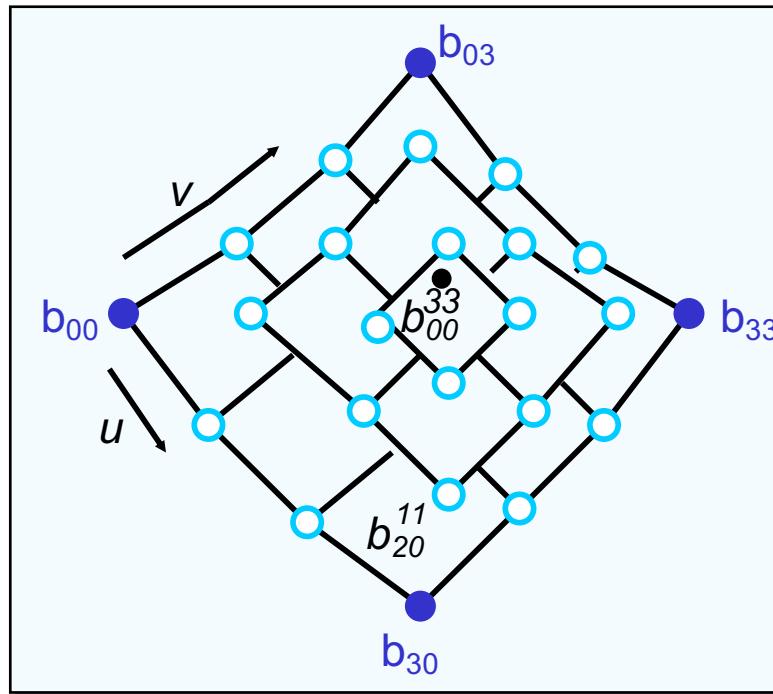
$$\begin{aligned}\mathbf{p}(u, v) = & (1-u) \cdot (1-v) \cdot \mathbf{p}_{00} \\ & + (1-u) \cdot (1-v) \cdot \mathbf{p}_{10} \\ & + (1-u) \cdot (1-v) \cdot \mathbf{p}_{01} \\ & + (1-u) \cdot (1-v) \cdot \mathbf{p}_{11}\end{aligned}$$

$$(u, v) \in [0, 1]^2$$



2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



*Der Algorithmus von de Casteljau für Flächen:
Der Punkt auf der Fläche wird durch wiederholte bilineare Interpolation berechnet.*



- **Die Beziertechnik hat 2 Nachteile:**

- keine lokale Kontrollierbarkeit

Die Veränderungen eines Bezierpunktes (oder eines Gewichtes bei der rationalen Form) verändern das ganze Segment.

- hoher Polynomgrad

Bei komplexen Formen wird eine größere Zahl von Bezier-Punkten benötigt. Mit jedem Bezier-Punkt erhöht sich der Grad um 1.

- Diese Nachteile werden mit der B-Spline-Technik ausgeschlossen.



■ **B-Spline-Technik**

B-Spline-Kurven verwenden als Basisfunktionen B-Spline-Funktionen.
Damit lassen sich folgende günstige Eigenschaften erreichen:

- Der Grad des Polynoms kann unabhängig von der Anzahl der Kontrollpunkte gewählt werden und
- Die Kontrollpunkte bestimmen den Verlauf der Kurve nur in einem lokalen Bereich der gesamten Kurve.

Eine **B-Spline-Funktion** der Ordnung k ist ein segmentweise definiertes Polynom vom Grad ($k - 1$), das an den Segmentübergängen im allgemeinen C^{k-2} stetig ist.

B-Spline-Funktionen lassen sich rekursiv definieren.

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



- Normalisierte B-Spline-Funktionen N_i^k der Ordnung k vom Grad $k-1$ lassen sich dann bei gegebenem Trägervektor $T = (t_0, t_1, \dots, t_{n+k})$ wie folgt definieren:

$$\text{Für } k=1 \quad N_i^1(t) = \begin{cases} 1 & \text{für } t_i \leq t < t_{i+1} \\ 0 & \text{sonst} \end{cases}$$

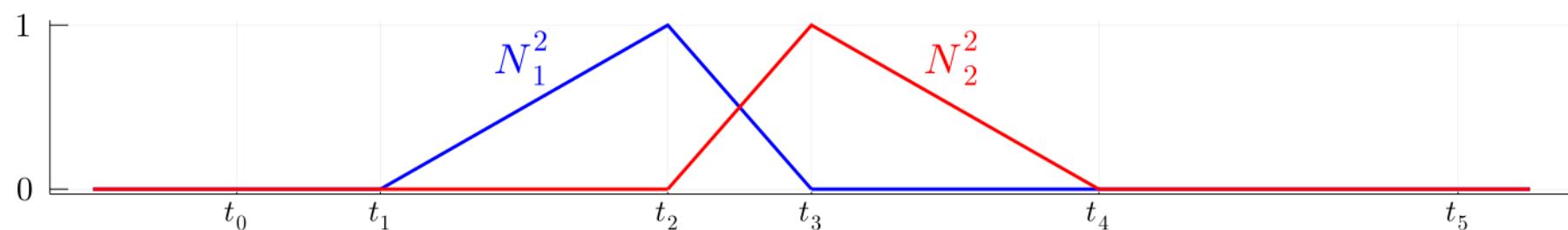
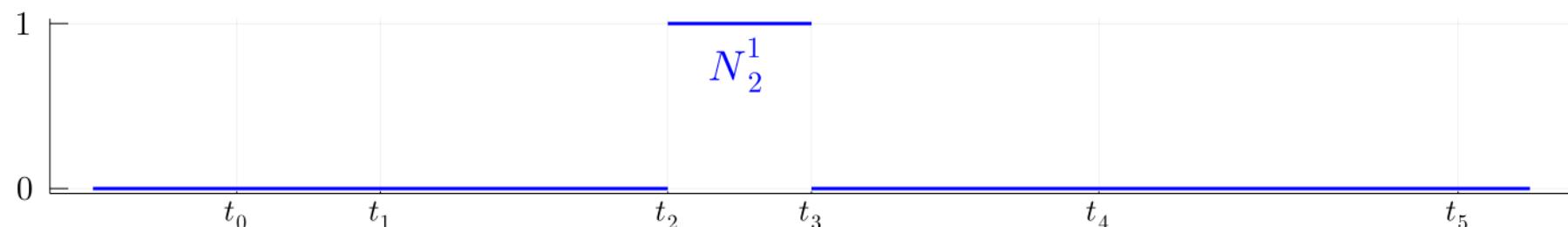
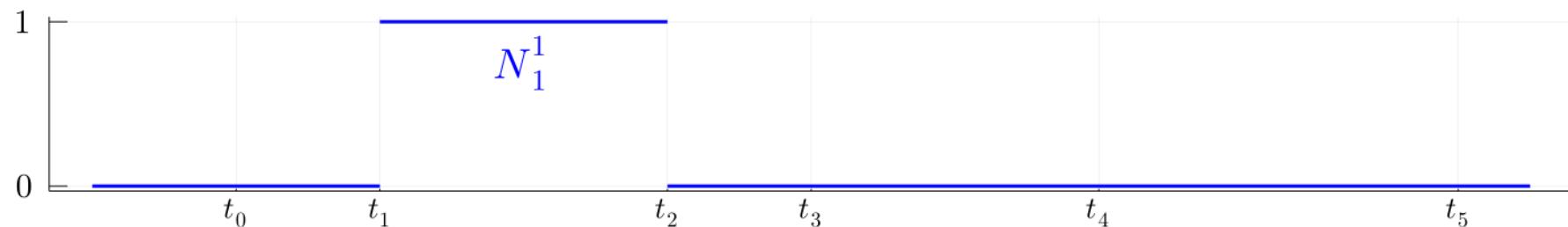
$$\text{für } k > 1 \quad N_i^k(t) = \frac{t - t_i}{t_{i+k-1} - t_i} N_i^{k-1}(t) + \frac{t_{i+k} - t}{t_{i+k} - t_{i+1}} N_{i+1}^{k-1}(t)$$

mit $i = 0, 1, \dots, n$

- Das heißt: B-Spline-Funktionen der Ordnung k werden aus B-Spline-Funktionen der Ordnung $k-1$ abgeleitet.

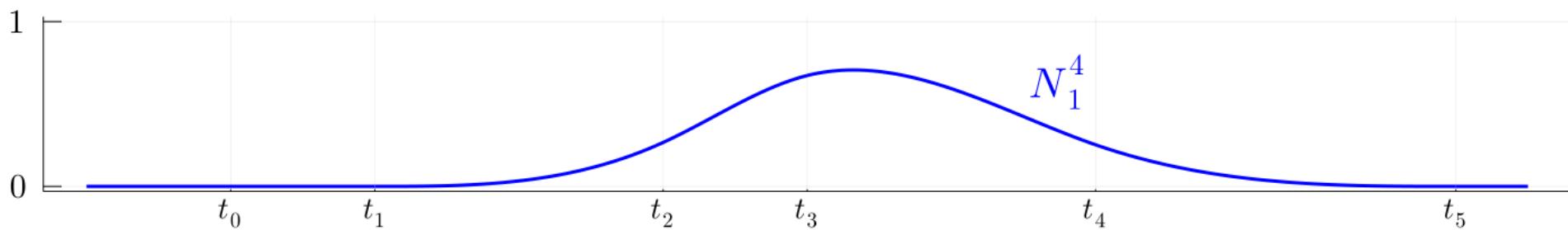
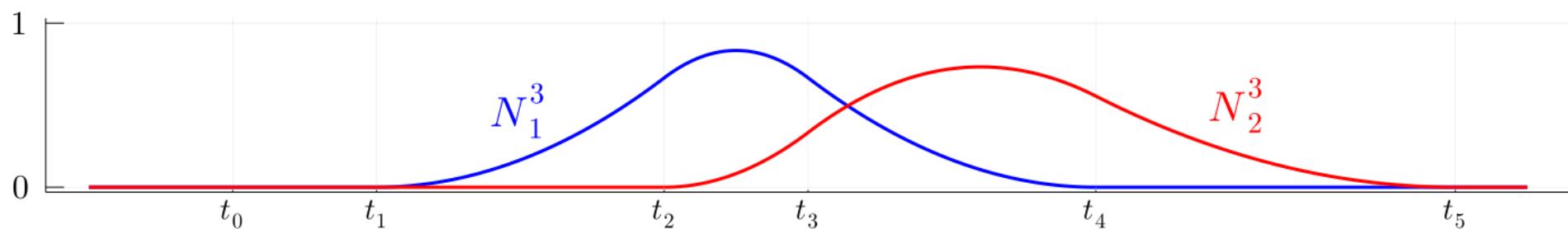
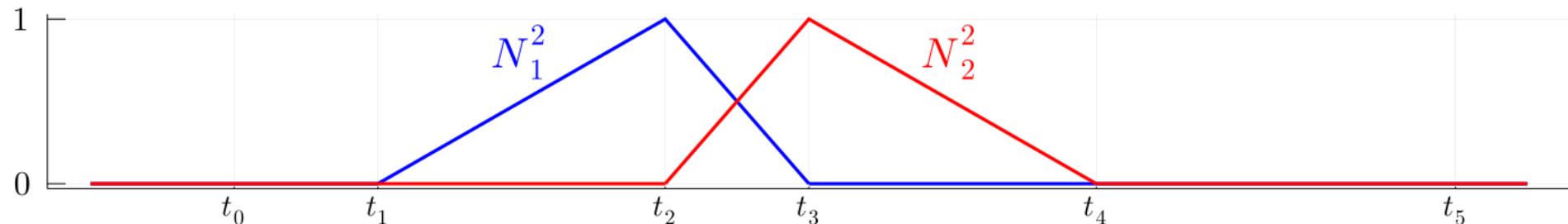
2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung





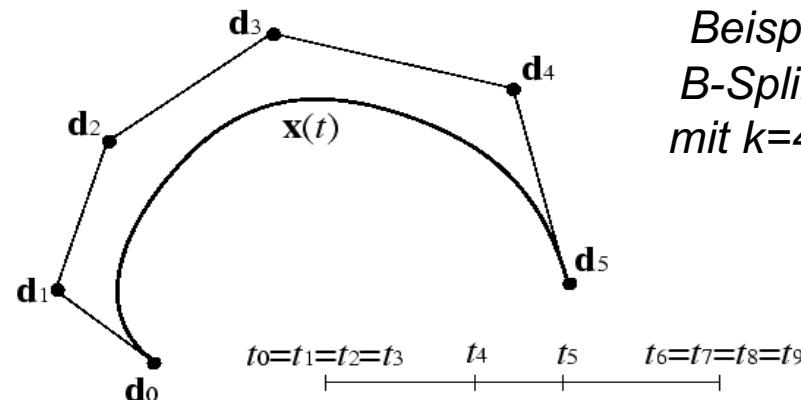
Zur Definition von B-Spline-Kurven können beliebig viele B-Spline-Funktionen gekoppelt werden, da diese nur über einem gewissen lokalen Träger wirken.

geg.:

- Kontrollpunkte (de Boor-Punkte) $\mathbf{d}_0, \dots, \mathbf{d}_n \in \mathbb{R}^2, \mathbb{R}^3$
- Trägervektor $T = (t_0, \dots, t_{n+k})$ mit $t_0 = \dots = t_{k-1}$ und
 $t_{n+1} = \dots = t_{n+k}$

Dann ist die B-Spline-Kurve der Ordnung k zum Träger T gegeben mit:

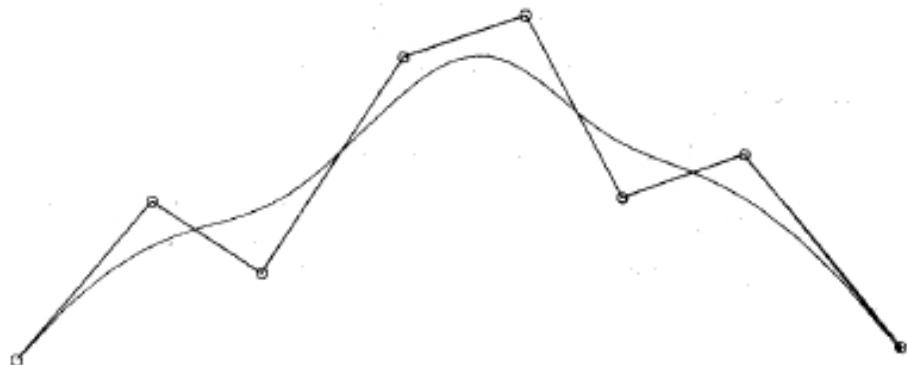
$$\mathbf{x}(t) = \sum_{i=0}^n \mathbf{d}_i * N_i^k(t)$$



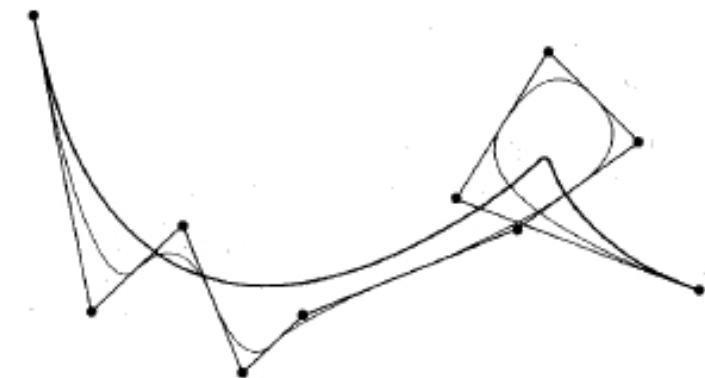
Beispiel einer
B-Spline-Kurve
mit $k=4$ und $n=5$

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



B-Spline-Kurve der Ordnung 4



B-Spline-Kurve der Ordnung 3

2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



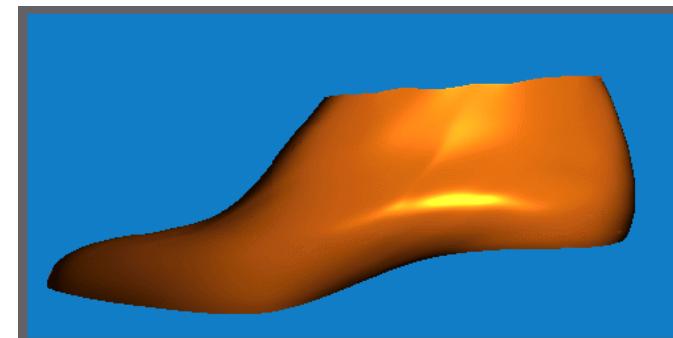
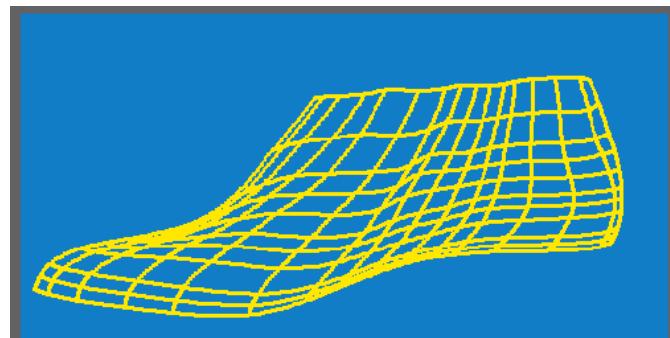
- Analog zur Definition der Kurven erfolgt die von Flächen:

Gegeben:

- Trägervektoren $U = (u_0, \dots, u_{n+k})$ und $V = (v_0, \dots, v_{m+l})$,
- de Boor Punkte \mathbf{d}_{ij} ($i = 0, \dots, n$; $j = 0, \dots, m$), bilden ein Kontrollnetz mit $(m+1) * (n+1)$ Kontrollpunkten.

Dann ist die B-Spline-Fläche gegeben mit:

$$\mathbf{x}(u, v) = \sum_{i=0}^n \sum_{j=0}^m \mathbf{d}_{ij} * N_i^k(u) * N_j^l(v)$$



2. Geometrische Modellierung

2.3 Kurven- und Flächenmodellierung



- Im Zusammenhang mit der Modellierung, insbesondere in CAD-Systemen, haben
NURBS (nichtuniformen rationalen B-Splines)
eine große Bedeutung.
- Die rationale B-Spline-Kurve wird analog zur rationalen Bezier-Kurve definiert als:

$$\mathbf{x}(t) = \frac{\sum_{i=0}^n w_i * \mathbf{d}_i * N_i^k(t)}{\sum_{i=0}^n w_i * N_i^k(t)}$$