

Computergraphik II

Winter 2012/2013

5 Modellierungstechniken

Versionsdatum: 9. November 2012



Prof. Dr. Andreas Kolb
Computer Graphics & Multimedia Systems

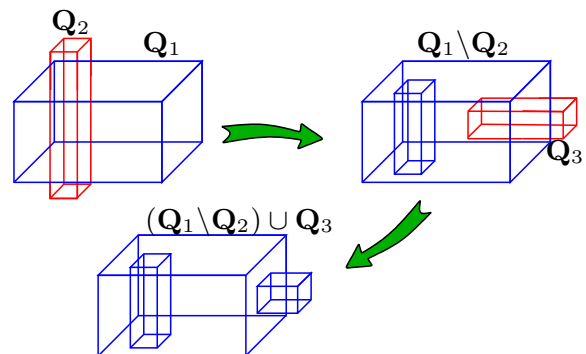
-Folie 5-1-

Computergraphik II

5.1 Solid Modeling



Motivation: Modellierung komplexer, geschlossener Körper (*Solids*) aus Primitiven unter Verwendung von Booleschen Operatoren.



Grundsätzliche Überlegungen zur Solid-Repräsentation

Modell-Repräsentation: Polygon, parametrisch, implizit, Subdivision

Eindeutigkeit: Zu jeder Geometrie sollte es nur eine Repräsentation geben

Repräsentationsformen: *Volumen-basiert* (alle Punkte **im** Objekt) oder *Oberflächen-basiert* (*boundary-representation, b-rep*) (Punkte **auf** Objektoberfläche)

Übliche Einschränkung: Polygonale b-reps



Prof. Dr. Andreas Kolb
Computer Graphics & Multimedia Systems

-Folie 5-2-

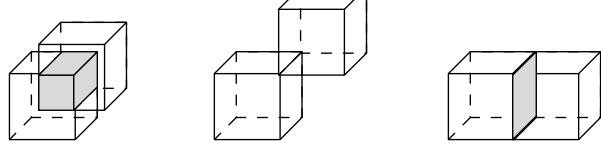
Computergraphik II

Boolsche Operatoren

Ziel: Einsatz boolscher Operationen zur Erzeugung komplexen Geometrien

Problem: Standard Boolsche Mengen-Operatoren erzeugen aus Solids u.U. keine Solids

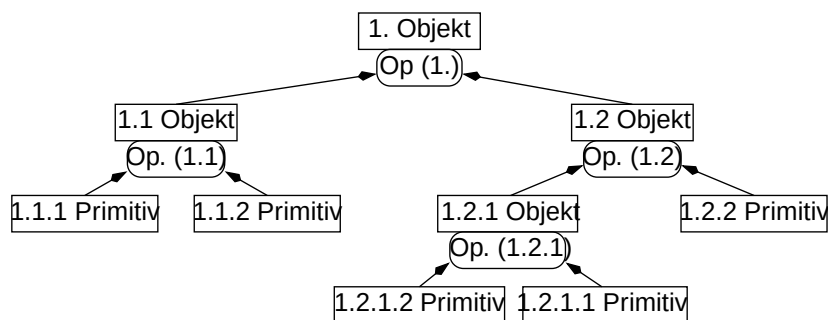
Bool'sche Schnittmenge:



Ergebnis: Solid Ergebnis: Linie Ergebnis: Fläche

Regularisierte boolsche Operatoren: Entfernen von Nicht-Solid Ergebnissen

Binärer Konstruktionsbaum: Sukzessive Anwendung binärer boolscher Operationen



5.1 Solid Modeling ...

Ansatz: Umsetzung Boolscher Operationen

Ziel: Ermittlung der b-rep einer Boolschen Operationen

Zerlegung in Teilpolygone: Auftrennen aller Polygone an den Schnittkanten mit den Polygonen des anderen Primitives

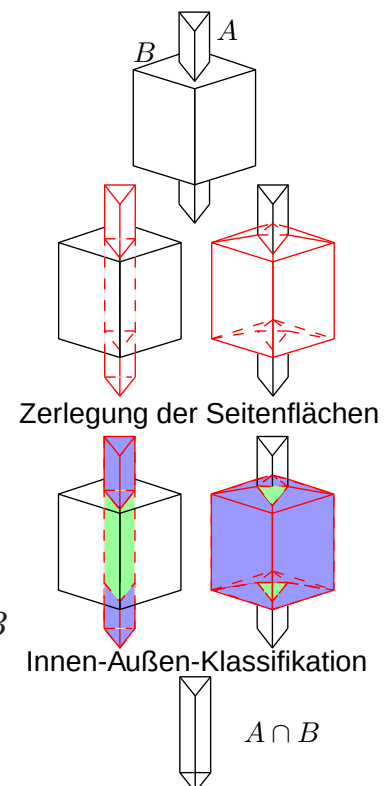
Innen-Außen-Klassifikation: Jedes entstandene Teilpolygone liegt bzgl. des anderen Primitives innen oder aussen

Zusammensetzung des Solids:

$A \cap B$: Alle Innen-Polygone von A und B

$A \cup B$: Alle Aussen-Polygone von A und B

$A \setminus B$: Alle Aussen-Polygone von A und alle Innen-Polygone von B



5.2 Fraktale Geometrie



Erzeugung komplexer, selbstähnlicher Geometrien (Berge, Pflanzen)

Erzeugungsprozess: Zur Erzeugung fraktaler Geometrien werden benötigt:

1. Ausgangselement aus einer Menge von Elementen
2. Rekursion (bis zu Rekursionstiefe): Ersetze aktuelles Element durch anderes (komplexeres) Element (Ersetzungsregel)
3. Geometrische Interpretation

Beispiel: Koch'sche Schneeflocke

Elementmenge: Polygonzüge

Ausgangselement: Polygonzug mit vier Kanten

Regel: ersetze jede Kante des aktuellen Polygons durch Ausgangselement



5.2 Fraktale Geometrie ...



Stochastische Fraktale

Stochastische Fraktale: Einsatz von Zufallsgrößen in Ersetzungsregel

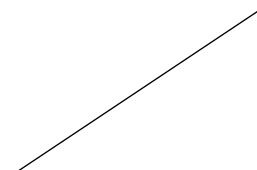
Fraktale Berge:

Elementmenge: Reguläre 3D-Dreiecksmeshes (Skizze!)

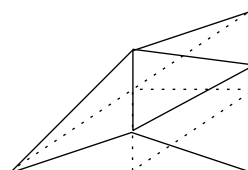
Ausgangselement: Ein beliebiges Dreieck in der $x - y$ -Ebene

Regel:

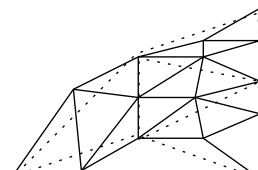
- Unterteile jedes Dreieck in vier ähnliche Dreiecke
- stochastische Störung der neuen Eckpunkte (z -Richtung)



Ausgangsdreieck



Erste Unterteilung



Zweite Unterteilung





Modellierung verzweigter Geometrien (z.B. Pflanzen)

Grundidee: Rekursive Anwendung von Ersetzungsregeln für geom. Primitive

Grundlage: Formale Sprachen ($\rightarrow$ EI-I):

Alphabet V : Beschreibt geom. Primitive, Transformationen & Stack-Befehle (Verzweigungen)

Wortmenge V^* : Menge aller Kombinationen aus Elementen aus V

Regeln: Für Elemente des Alphabets: $a \mapsto p(a) \in V^*$, $\forall a \in V$ bzw. für Worte:

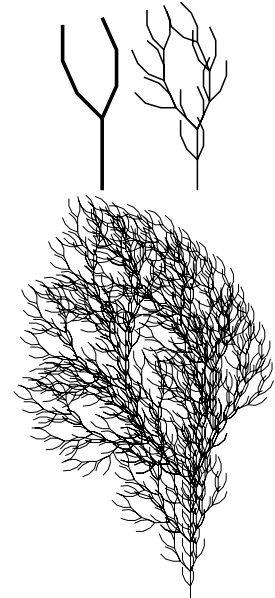
$$\omega = a_1 a_2 a_3 \dots \mapsto p(\omega) = p(a_1) p(a_2) p(a_3) \dots$$

Axiom: Ausgangselement $\omega_0 \in V^*$ der Rekursion

Beispiel: Alphabet bestehend aus

a zeichne Linie (initiale Richtung: oben)
 $+$, $-$ Richtungswechsel um $\pm 22.5^\circ$
 $[$, $]$ Stackingbefehl (Push- und Pop-Stack)

Axiom a und Regel $p(a) = aa + [+a - a - a] - [-a + a + a]$, sonst $p(x) = x$



5.3 Lindenmayer-Systeme ...



Geometrische Interpretation: Elemente des Alphabetes stehen für geometrische Primitive, Transformationen oder Stack-Befehle

Algorithmus konkret zum vorherigen Beispiel:

```
void InterpretelSystem ( char* word, Point P, Vector d) {
    moveto(P);          // Startposition
    while ( (*word) != EndOfWord ) {
        switch ( *word ) {
            case 'a':    P = P + d;      lineto(P);      break;
            case '+':    d = rotate(d, phi);             break;
            case '-':    d = rotate(d, -phi);            break;
            case '[':    InterpretelSystem(word+1, P, d);
                        moveto(P);          // Zeichenpos. VOR '['
                        word = FindMatchingBrace(word); // ueberspringe [...]
                        break;
            case ']':    return;
        }
        word ++;        // naechster Buchstabe im Wort
    }
}
```



5.4 Kurvenbasierte Techniken



Erinnerung: TP-Flächen verallgemeinern ein Kurvenschema

Problem: Angabe von $n \cdot m$ Kontrollpunkten unhandlich/nicht problemorientiert

Sweep-Fläche: Alternativer Kurvenansatz:

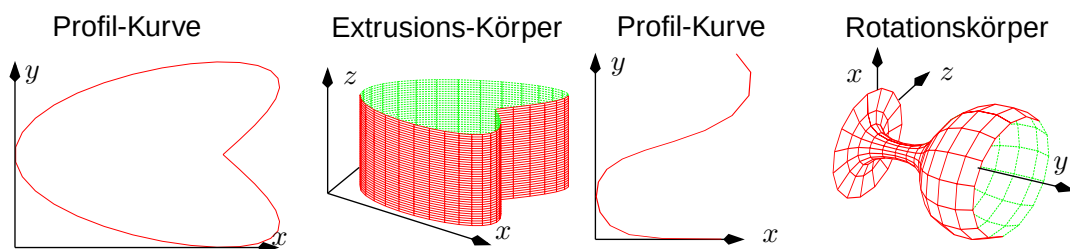
Kurven: Bewege *Profil-Kurve* $S(u)$ entlang eines *Pfades* $C(v)$.

Variante: Veränderung der Profil-Kurve S in Abhängigkeit von v

Ausprägungen: Konkrete Kurven geben spezielle Klassen von Flächen:

Rotationskörper: Der Pfad C ist ein Kreis

Extrusionskörper: Geschlossenes 2D-Profil S ; Pfad C ist häufig eine Linie.



5.4 Kurvenbasierte Techniken ...



Beispiel: Einfacher Extrusionskörper

Ziel: Bestimme Polygon-Approximation des herzförmigen Extrusionskörpers

Profil-Kurve $S(u)$, $u \in [0, 1]$ und **Pfad-Kurve** $C(v)$, $v \in [0, 1]$ gegeben durch

$$S(u) = \begin{pmatrix} s_x(u) \\ s_y(u) \end{pmatrix} = \begin{pmatrix} \cos(2\pi u) \cdot \sin(\pi u) \\ \cos(2\pi u) \end{pmatrix}, \quad C(v) = \begin{pmatrix} 0 \\ 0 \\ v \end{pmatrix}$$

Parametrische Darstellung: $F(u, v) = \begin{pmatrix} s_x(u) \\ s_y(u) \\ v \end{pmatrix}, \quad (u, v) \in [0, 1]^2$

Polygonalisierung:

○ Gitter im Parametergebiet $[0, 1]^2$ mit Auflösung $(N_u + 1) \times (N_v + 1)$:

$$P_{ij} = \begin{pmatrix} i\Delta_u \\ j\Delta_v \end{pmatrix}, \quad i \in \{0, \dots, N_u\}, \quad j \in \{0, \dots, N_v\} \text{ wobei } \Delta_u = \frac{1}{N_u}, \quad \Delta_v = \frac{1}{N_v}$$

○ Polygone ergeben sich zu dann zu:

$$\square(F(P_{i,j}), F(P_{i+1,j}), F(P_{i+1,j+1}), F(P_{i,j+1}))$$



Allgemeine Extrusionskörper

Ausgangslage: Allgemeine Extrusionsfläche sind gegeben durch

2D-Profil: Ebene, geschlossene Kurven

$$S_i : u \mapsto S_i(u) \in \mathbb{R}^2, u \in [0, 1], i = 1, \dots, n$$

Pfad: Kurve, entlang derer die Profilkurven S_i aufgereiht werden:

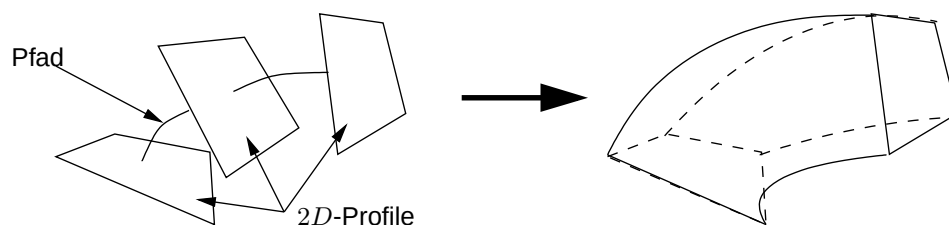
$$C : v \mapsto C(v) \in \mathbb{R}^3, v \in [0, 1]$$

Zuordnung: Für jedes S_i werden zusätzlich benötigt:

- ein Ausrichtungspunkt M_i (z.B. Schwerpunkt)
- ein Pfad-Parameter $v_i \in [0, 1]$ mit $M_i \hat{=} C(v_i)$ und $v_{i-1} < v_i < v_{i+1}$

Arbeitsschritte: 1. Ausrichtung der Profile am Pfad (*Frenetsches Dreibein*)

2. Interpolation zwischen den Profilen (z.B. B-Splines)



5.4 Kurvenbasierte Techniken ...



Algorithmus: Polygon-Approximation des Extrusionsobjektes

Gegeben: 1. Pfad-Kurve: $C : v \mapsto C(v) \in \mathbb{R}^3, v \in [0, 1]$

2. Ebene 2D-Profil-Kurven $S_i : u \mapsto S_i(u) \in \mathbb{R}^2, u \in [0, 1], i = 1, \dots, n$ mit

2.1. Ausrichtungspunkt M_i

2.2. Pfad-Parameter $v_i \in [0, 1]$ mit $M_i \hat{=} C(v_i)$

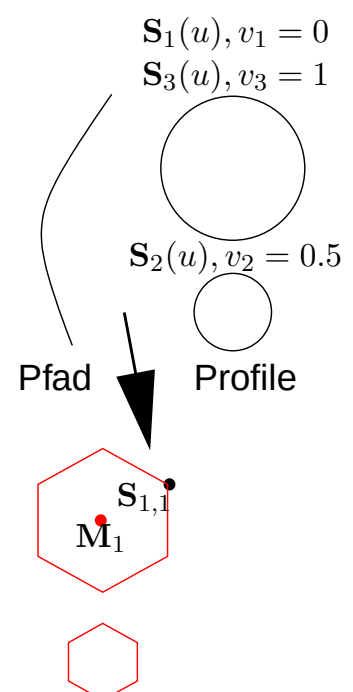
2.3. Auflösung für den Pfad r_C und Profile r_S

Linearisierung der Profile: Approximation der Profile mit r_S Punkten bzgl. Schwerpunkt M_i :

$$S_{i,j} = S_i(u_j) - M_i \in \mathbb{R}^2$$

$$\text{mit } u_j = \frac{j}{r_S}$$

M_i liegt damit im Ursprung



Algorithmus: Polygon-Approximation des Extrusionsobjektes (Forts.)

Einpassen der Polygone für Profil S_i am Frenet'schen Dreibein in $C(v_i)$:

$$S_{i,j} = \begin{pmatrix} x_{i,j} \\ y_{i,j} \end{pmatrix} \mapsto S_{i,j}^* \text{ mit}$$

$$S_{i,j}^* = C(v_i) + x_{i,j} \hat{n}_C(v_i) + y_{i,j} \hat{b}_C(v_i) \in \mathbb{R}^3$$

B-Spline Interpolation der Punkte $S_{i,j}^*, i = 1, \dots, n$ durch Kurve $S_j^*(v)$ (B-Spline, Catmull-Rom)

Linearisierung der B-Splines durch Polygonzüge bestehend aus r_C Strecken

