

Computergraphik II

Prof. Dr. Andreas Kolb

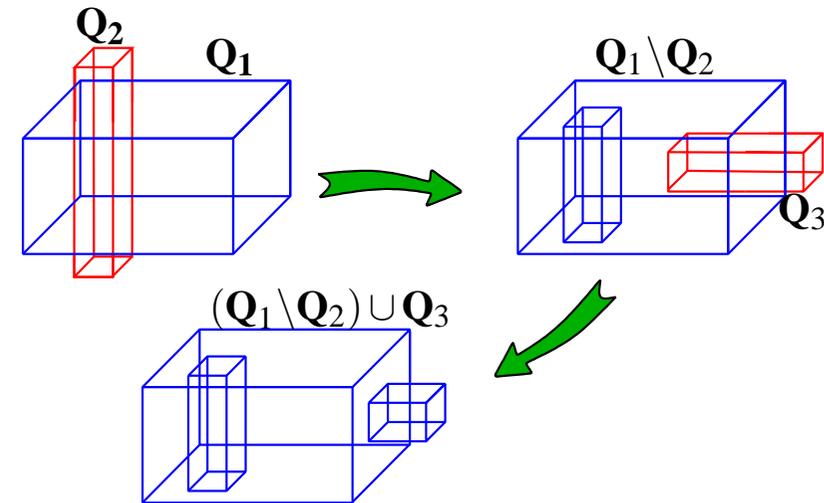
Fachgruppe Computergraphik und Multimediasysteme

Universität Siegen – Fachbereich 12

Version: 8. November 2004

5.1 Solid Modeling

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



Grundsätzliche Überlegungen zur Solid-Repräsentation

Darstellungsraum: Objekt-Art (technisch, Animation, ...) und 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

5 Modellierungstechniken

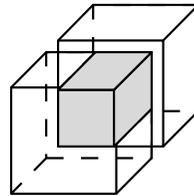


Boolsche Operatoren

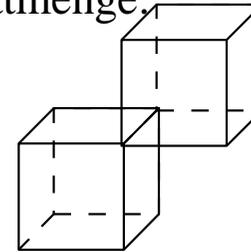
Ziel: Einsatz boolscher Operationen zur Erzeugung komplexen Geometrien

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

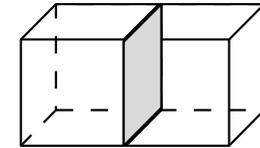
Bool'sche Schnittmenge:



Ergebnis: Solid



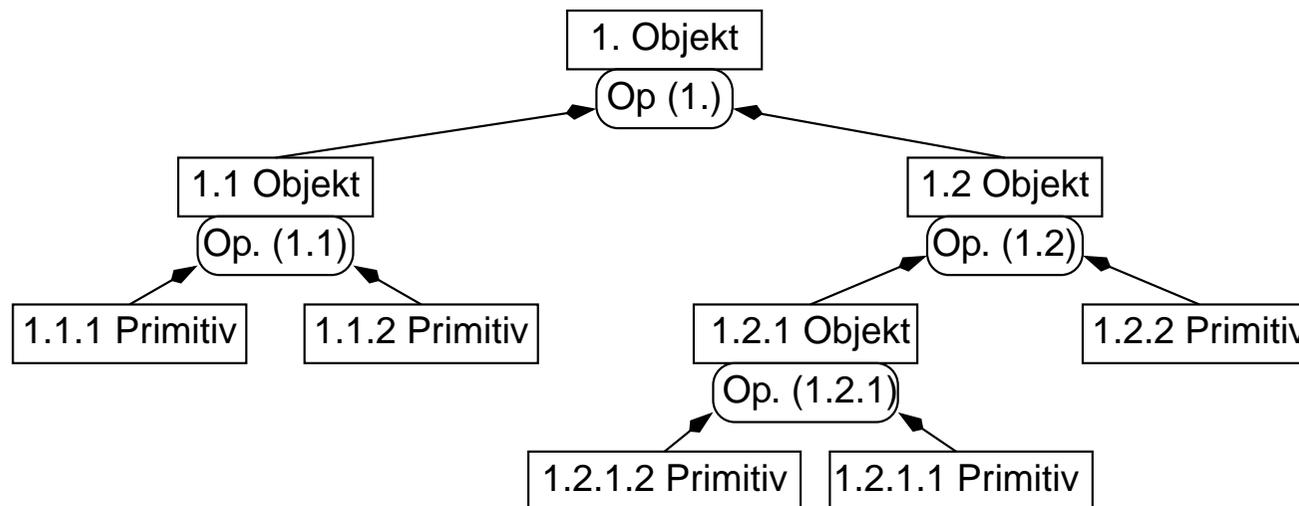
Ergebnis: Linie



Ergebnis: Fläche

Regularisierte boolsche Operatoren: Entfernen von Nicht-Solid Ergebnissen

Binärer Konstruktionsbaum: Sukzessive Anwendung binärer boolscher Operationen



Ansatz: Umsetzung Boolescher Operationen

Ziel: Ermittlung der b-rep einer Booleschen Operationen

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

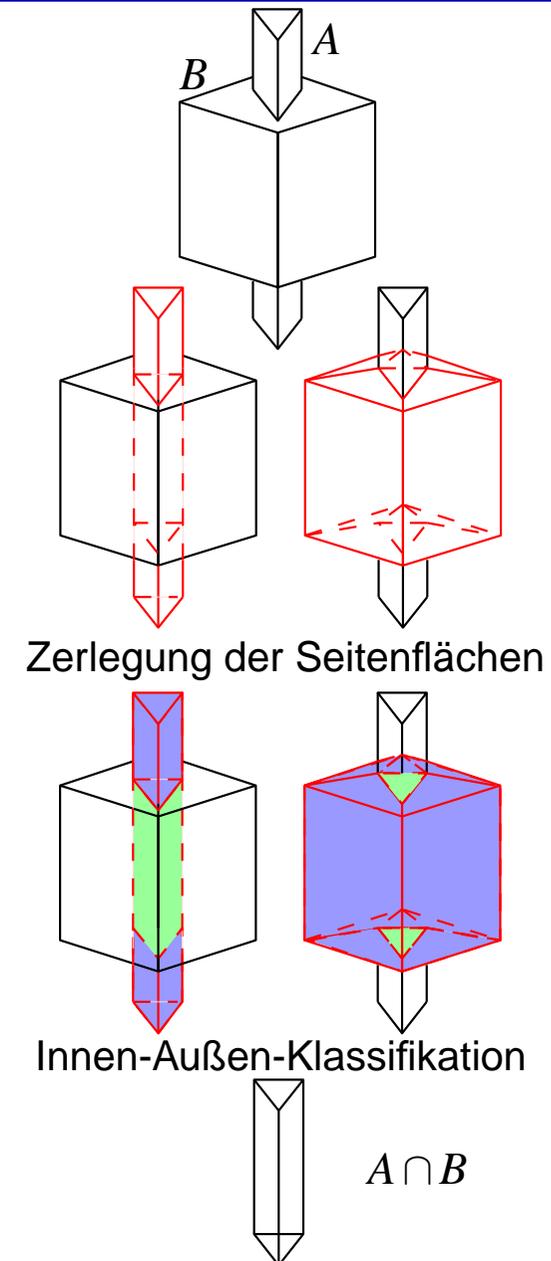
Innen-Außen-Klassifikation: Jedes entstandene Teilpolygone liegt bzgl. des anderen Primitives innen oder aussen

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) durch fraktale Prozesse

Erzeugungsprozeß: Bei der Erzeugung von fraktalen Geometrien werden benötigt:

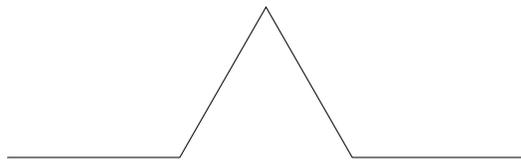
1. Ausgangselement aus einer Menge von Elementen
2. Rekursion (bis zu Rekursionstiefe): Ersetze aktuelles Element durch anderes (komplexeres) Element (Ersetzungsregel)
3. Geometrische Interpretation

Beispiel: Koch'sche Schneeflocke

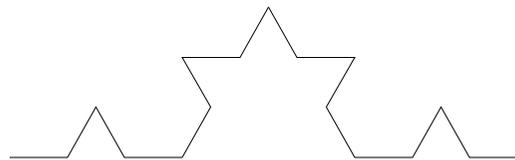
Elementmenge: Polygonzüge

Ausgangselement: Polygonzug mit vier Kanten

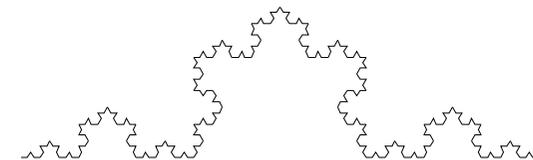
Regel: ersetze jede Kante des aktuellen Polygons durch Ausgangselement



Ausgangselement



1. Rekursion



4. Rekursion

Stochastische Fraktale

Stochastische Fraktale: Einsatz von Zufallsgrößen in Ersetzungsregel

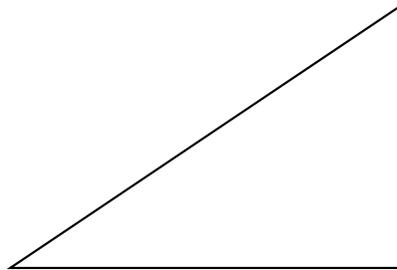
Fraktale Berge:

Elementmenge: Reguläre $3D$ -Dreiecksmeshes (Skizze!)

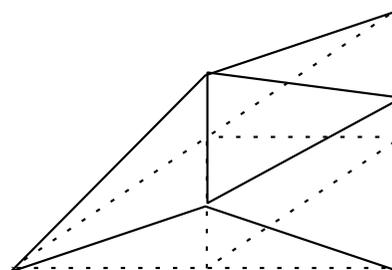
Ausgangselement: Ein beliebiges Dreieck in der $x - y$ -Ebene

Regel:

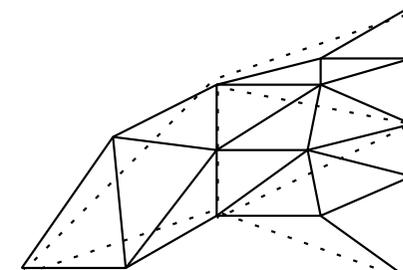
- Unterteile jedes Dreieck in vier ähnliche Dreiecke
- stochastische Störung der neuen Eckpunkte (z -Richtung)



Ausgangsdreieck



Erste Unterteilung



Zweite Unterteilung

5.3 Lindenmayer-Systeme



Ziel: Modellierung verzweigter Geometrien (Pflanzen), Simulation von Wachstumsprozessen
Lindenmayer-Systeme (L-Systeme): Fraktale Struktur mit

Elementmenge: Wortmenge V^* basierend auf Alphabet V

V^* : Menge aller Kombinationen aus Elementen aus V

Bsp.: $V = \{a, +, -\}$, Elemente aus V^* : $111, 1a1a\$, \dots$

Ausgangselement: $\omega_0 \in V^*$ (Axiom)

Regel: Für Elemente des Alphabets: $a \mapsto p(a) \in V^*$, $\forall a \in V$

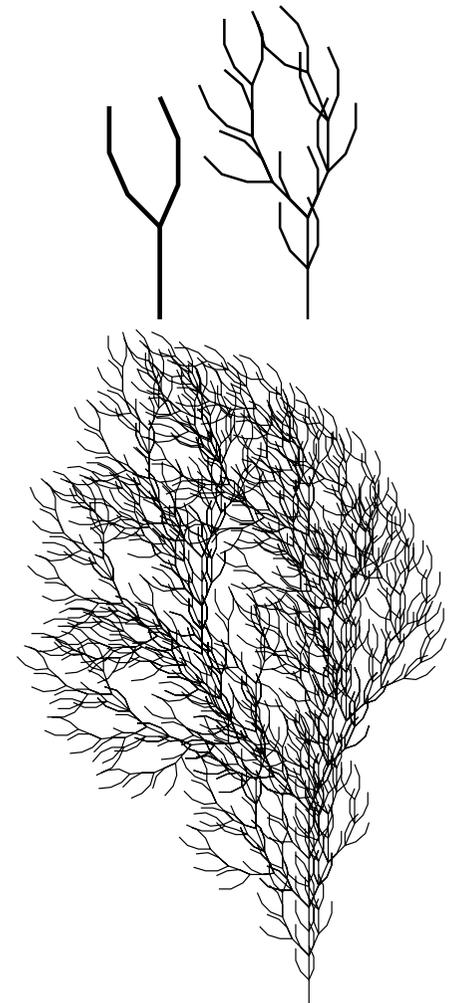
Wort: $\omega = a_1a_2a_3 \dots \mapsto p(\omega) = p(a_1)p(a_2)p(a_3)\dots$

Beispiel: Alphabet: $\{a, +, -, [,]\}$, **Axiom:** a

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

Geometrische Interpretation:

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





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 , 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:

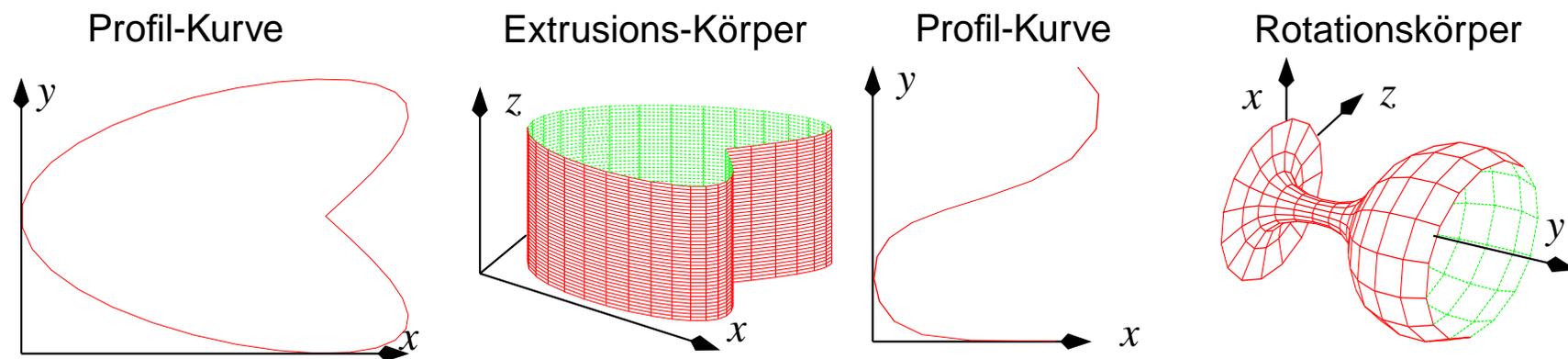
Kurven: Bewege *Profil-Kurve* $S(u)$ entlang eines *Pfades* $C(v)$.

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: Errechnung der Polygon-Approximation des herzförmigen Extrusionskörpers

Kurven: Profil-Kurve $\mathbf{S}(u)$, $u \in [0, 1]$ und Pfad-Kurve $\mathbf{C}(v)$, $v \in [0, 1]$ gegeben durch

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

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

Polygonalisierung:

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

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

- Polygone ergeben sich zu dann zu: $\square (\mathbf{F}(\mathbf{P}_{i,j}), \mathbf{F}(\mathbf{P}_{i+1,j}), \mathbf{F}(\mathbf{P}_{i+1,j+1}), \mathbf{F}(\mathbf{P}_{i,j+1}))$



Allgemeine Extrusionskörper

Ausgangslage: Allgemeine Extrusionsfläche sind gegeben durch

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

Pfad: Kurve, entlang derer die Profilkurven \mathbf{S}_i aufgereiht werden:

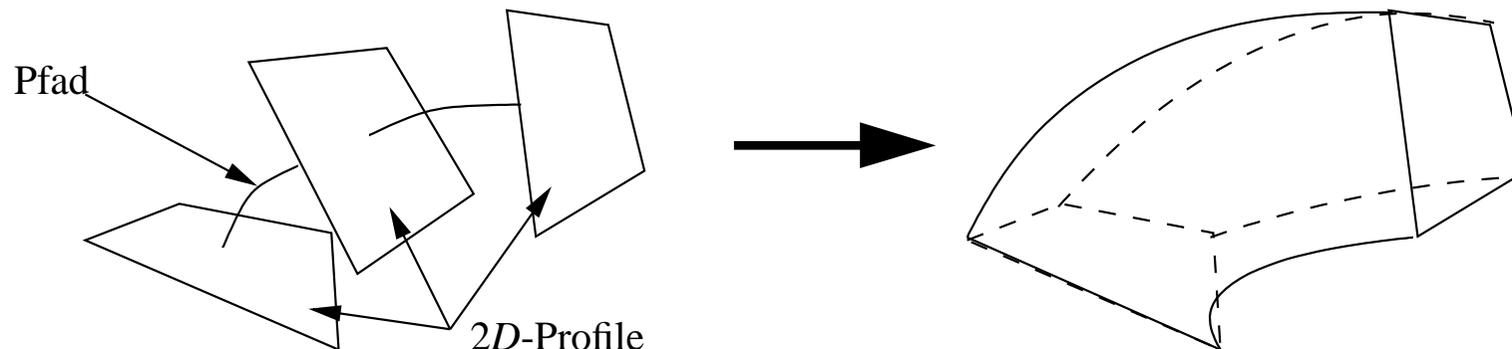
$$\mathbf{C} : v \mapsto \mathbf{C}(v) \in \mathbb{R}^3, v \in [0, 1]$$

Zuordnung: Für jedes \mathbf{S}_i werden zusätzlich benötigt:

- ein Ausrichtungspunkt \mathbf{M}_i (z.B. Schwerpunkt)
- ein Pfad-Parameter $v_i \in [0, 1]$ mit $\mathbf{M}_i \hat{=} \mathbf{C}(v_i)$ 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

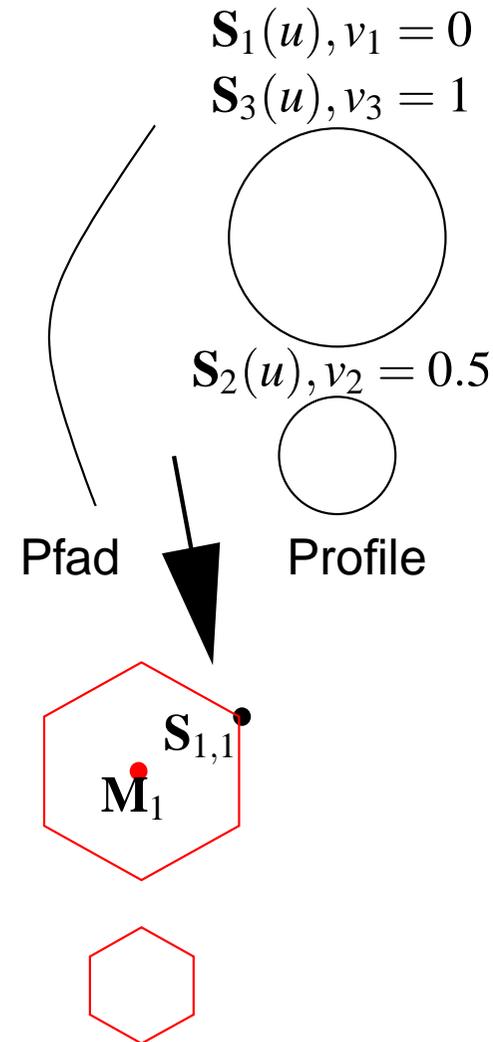
- Gegeben:**
1. Pfad-Kurve: $\mathbf{C} : v \mapsto \mathbf{C}(v) \in \mathbb{R}^3, v \in [0, 1]$
 2. Ebene 2D-Profil-Kurven $\mathbf{S}_i : u \mapsto \mathbf{S}_i(u) \in \mathbb{R}^2, u \in [0, 1], i = 1, \dots, n$ mit
 - 2.1. Ausrichtungspunkt \mathbf{M}_i
 - 2.2. Pfad-Parameter $v_i \in [0, 1]$ mit $\mathbf{M}_i \hat{=} \mathbf{C}(v_i)$
 - 2.3. Auflösung für den Pfad r_C und Profile r_S

Linearisierung der Profile: Approximation der Profile mit r_S Punkten bzgl. Schwerpunkt \mathbf{M}_i :

$$\mathbf{S}_{i,j} = \mathbf{S}_i(u_j) - \mathbf{M}_i \in \mathbb{R}^2$$

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

\mathbf{M}_i liegt damit im Ursprung



Algorithmus: Polygon-Approximation des Extrusionsobjektes (Forts.)

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

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

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

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

