Computer Graphics II
6: Modeling Techniques

Computer Graphics and
Multimedia Systems Group

University of Siegen

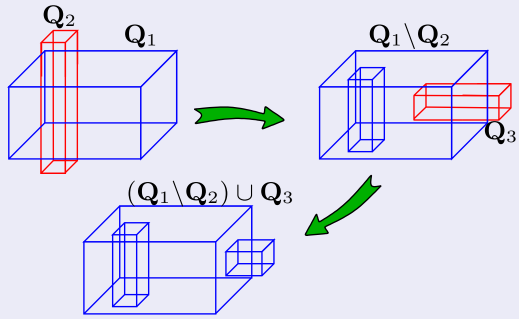
Structure of Chapter

- *Subsection 1: Solid Modeling, i.e. defining compelx shapes using boolean operations for solids (volumetric objects)*
- *Subsection 2: Generation of fractal geometry by recursion processess*
- *Subsection 3: Extended version of fractals using Lindenmayer systems*
- *Subsection 4: Shape definitions using curve methods*

Motivation (General Considerations)

So far: *Model representation based on polygons, parametric, implicite or subdivision surfaces.*

Idea: *Modeling of complex,* closed *bodies (solids) from primitives using boolean operators.*



Uniqueness Requirement: *For each geometry there should only be one representation*

Solid representations *can be volume based (all points* **inside** *the object) or surface-based (boundary-representation, b-rep)*

Example: *Surface vs. volume based representation of a sphere*

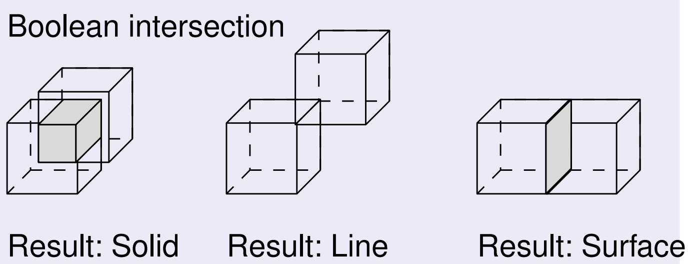
$$\mathbf{S}(u,v) = \begin{pmatrix} \cos u \cos v \\ \cos u \sin v \\ \sin u \end{pmatrix}, \qquad \mathbf{S}(u,v,w) = \begin{pmatrix} w \cos u \cos v \\ w \cos u \sin v \\ w \sin u \end{pmatrix}, \quad \begin{matrix} u \in [-\frac{\pi}{2}, \frac{\pi}{2}] \\ v \in [-\pi, \pi] \\ w \in [0,1] \end{matrix}$$

Most Common Approach: *Polygonal b-reps*

Approach

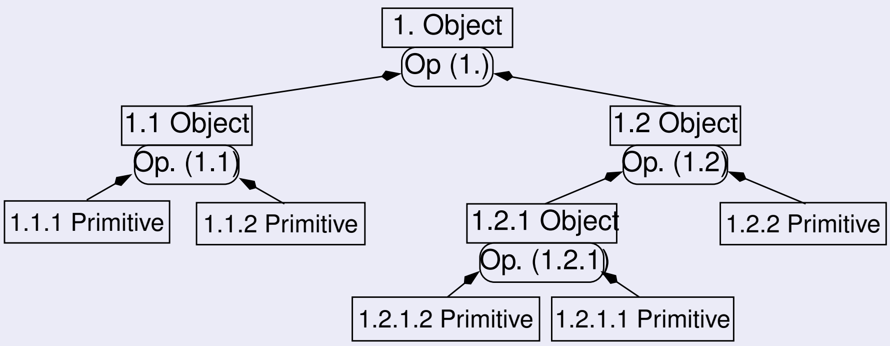
Goal: *Use of Boolean operators for the creation of complex geometries*

Problem: *Standard boolean set operators do not necessarily create solids from solids*



Regularized Boolean Operators: *Removal of non-solid results*

Binary Construction Tree: *Successive application of binary Boolean operations*



Approach (Implementation of Boolean Operations)

Goal: *Determination of b-rep after application of a boolean operation*

Segment in Partial Polygons: *Cut all polgons at the intersection lines to the polygons of other object*

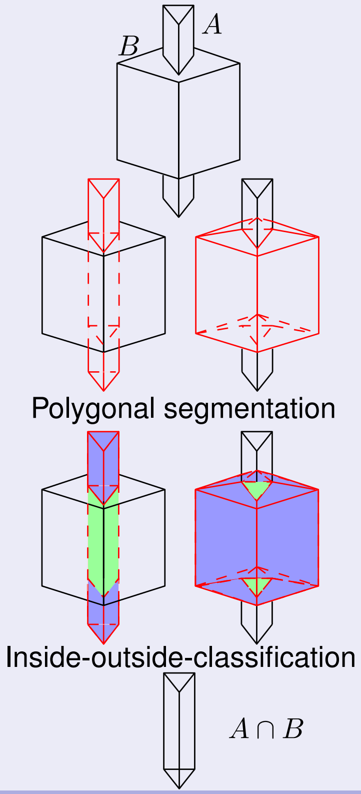
Inside-Outside-Classification: *Each resulting partial polygon lies inside or outside with reference to the other primitive*

Composition of the Solid:

$A \bigcap B$: *All inside-polygons of A and B*

$A \bigcup B$: *All outside-polygons of A and B*

$A \backslash B$: *All outside-polygons of A and all inside-polygons of B*



Approach

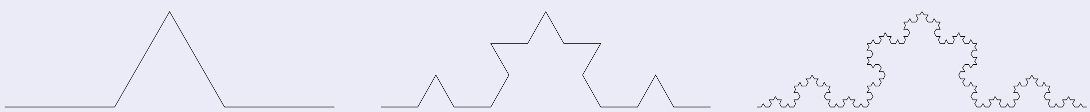
Generation *of complex, self-similar geometry (hills, plants)*

Requirements *for the generation of fractal geometry:*

1. *Initial Element from a Set of Elements*
2. *Recursion: Replace current element with another (more complex) element using replacement rules*
3. *Geometric interpretation*

Example: *Koch Snowflake*

Element Set: *Here, polygonal strips*
Initial Element: *Polygonal chain with four edges*
Rule: *replace each edge of the current polygon with the initial element*



initial element *1. recursion* *4. recursion*

Stochastic Fractals

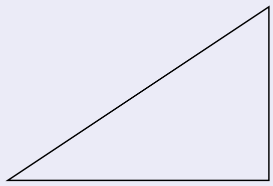
Approach

Stochastic Fractals: *Application of random variables in replacement rule*

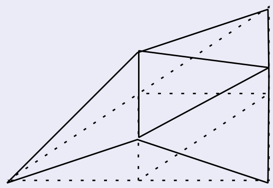
Fractal Mountains:

Set of Elements: *Regular $3D$-triangular meshes (sketch!)*
Initial Element: *A random triangle in the $x-y-$plain*
Rule:

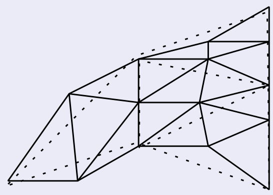
- *Divide each triangle into four similar triangles*
- *Stochastic perturbance of the new vertices, mainly in z-direction*



initial triangle



first subdivision



second subdivision

6.3: Lindenmeyer Systems (L-Systems)

Approach (Grammar-Based Fractals)

Idea: *Use grammar-like recursive application of replacement rules*

Basis: *Formal languages (see lecture on Algorithm & Data Structures):*

Alphabet V: *Describes geom. primitives, transformations & stack op. (used for branching)*

Words V^*: *Set of combinations of symbols from V*

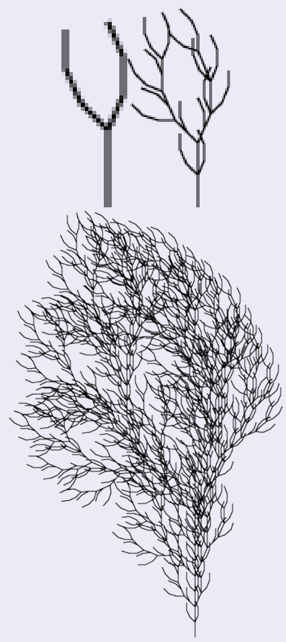
Rules: *For elements of the alphabet:*
$a \mapsto p(a) \in V^*,\ \forall a \in V$ *or for words:*
$\omega = a_1 a_2 a_3 \ldots \mapsto p(\omega) = p(a_1)p(a_2)p(a_3)\ldots$

Axiom: *Initial element $\omega_0 \in V^*$ of the recursion*

Example: *Alphabet consisting of:*

a	*draw line (initial direction: up)*
$+,-$	*change of direction by $\pm 22.5°$*
$[,]$	*stack command (push- and pop)*

Axiom a and rule $p(a) = aa+[+a-a-a]-[-a+a+a]$, otherwise $p(x) = x$



Algorithm (Interpretation of L-Systems)

Geometric Interpretation: *Elements of the alphabet represent geometric primitives, transformations or stack instructions*

Algorithm *specifically for the previous example:*

```
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);
        word = FindMatchingBrace(word); // jump to resp. ']'
        moveto(P);                      // last draw pos. before '['
        break;
      case ']':  return;
    }
    word ++;            // go to next symbol
  }
}
```


Approach

Reminder: *TP-surfaces generalize a curve scheme*

Problem: *Declaration of $n \cdot m$ control points is a time-consuming task*

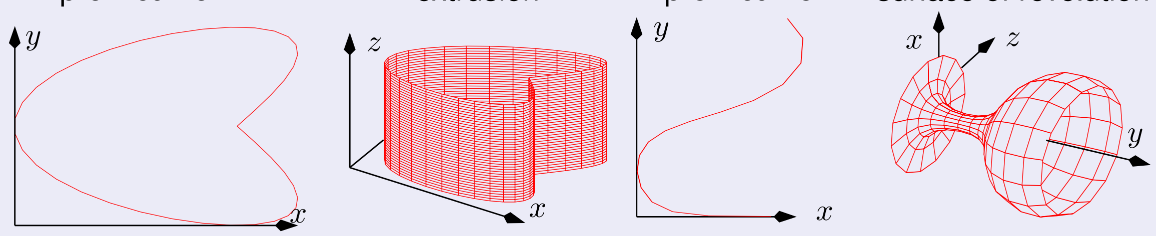
Sweep-surface: *Alternative curve approach:*

Curves: *Move profil-curve $\mathbf{S}(u)$ along a path $\mathbf{C}(v)$.*
Alternative: Variation of the profile-curve $\mathbf{S}$ depending on v

Example: *Certain curves produce specific classes of surfaces:*

Surface of revolution: *The path $\mathbf{C}$ is a circle*
Extrusion: *Closed 2D-profile $\mathbf{S}$; path $\mathbf{C}$ is often a line.*

profil curve — extrusion — profil curve — surface of revolution



6.4: Curve-Based Techniques

Example (A Simple Extrusion Shape)

Goal: Polygon approximation of a heart-shaped extrusion shape.

Profile-curve $\mathbf{S}(u),\ u \in [0,1]$ and path-curve $\mathbf{C}(v),\ v \in [0,1]$ given by

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

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

Polygonalization:

- Mesh in the areas of parameters $[0,1]^2$ with a resolution $(N_u+1) \times (N_v+1)$:

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

- Polygons are given by: $\Box\left(\mathbf{F}(\mathbf{P}_{i,j}), \mathbf{F}(\mathbf{P}_{i+1,j}), \mathbf{F}(\mathbf{P}_{i+1,j+1}), \mathbf{F}(\mathbf{P}_{i,j+1})\right)$

General Extrusion Shapes

Approach

Definition: *A general extrusion surface is given by*

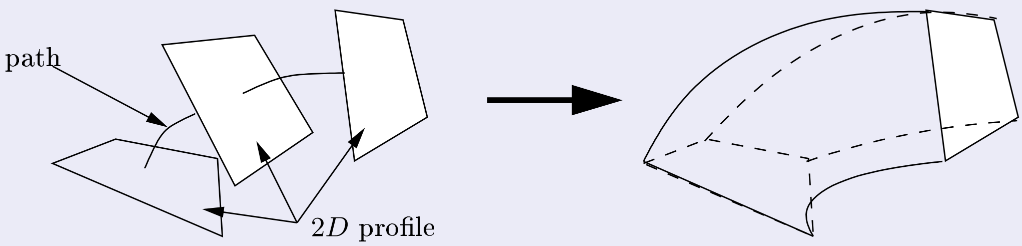
$2D$-profile: *Plain, closed curves* $\mathbf{S}_i : u \mapsto \mathbf{S}_i(u) \in \mathbb{R}^2,\ u \in [0,1],\ \ i = 1,\ldots,n$

Path: *curve, along which the profile curves* $\mathbf{S}_i$ *are located:*
$\mathbf{C} : v \mapsto \mathbf{C}(v) \in \mathbb{R}^3,\ v \in [0,1]$

Mapping: *For each* $\mathbf{S}_i$ *we addionally need:*

- *one focal point* $\mathbf{M}_i$
- *one path parameter* $v_i \in [0,1]$ *with* $\mathbf{M}_i \hat{=} \mathbf{C}(v_i)$ *and* $v_{i-1} < v_i < v_{i+1}$

Procedure: 1 *Alignment of the profiles to the path (Frenet-Serret frame)*
2 *Interpolation between the profiles (e.g. Catmul-Rom splines)*



Algorithm (Polygon-Approximation of an Extrusion Surface)

Given: ❶ *Path-curve:* $\mathbf{C} : v \mapsto \mathbf{C}(v) \in \mathbb{R}^3,\ v \in [0,1]$

❷ *Plain $2D$-profile-curves* $\mathbf{S}_i : u \mapsto \mathbf{S}_i(u) \in \mathbb{R}^2,\ u \in [0,1],\ i = 1,\dots,n$ *with*

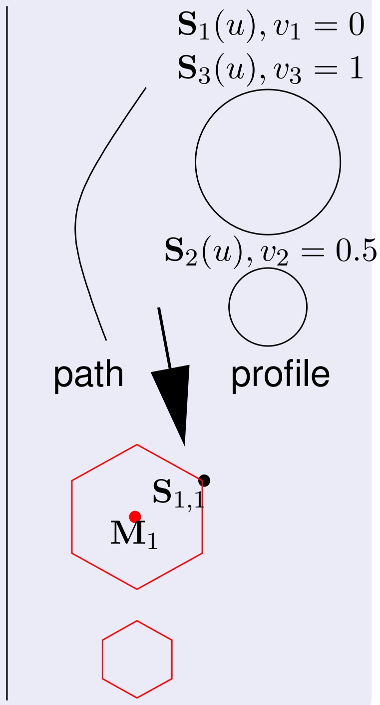
- *Alignment point* $\mathbf{M}_i$
- *Path-Parameter* $v_i \in [0,1]$ *with* $\mathbf{M}_i \hat{=} \mathbf{C}(v_i)$
- *Resolution for path* $r_\mathbf{C}$ *and profiles* $r_\mathbf{S}$

Profile Linearization: *Approximation of profiles with $r_\mathbf{S}$ points with reference to focal point $\mathbf{M}_i$:*

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

$$\text{with } u_j = \frac{j}{r_\mathbf{S}}$$

$\mathbf{M}_i$ *is therefore located at the origin*



Algorithm (Polygon-Approximation of the Extrusion Surface (Cont.))

Fitting of the polygons *for profile $\mathbf{S}_i$ using the Frenet-Serret frame in $\mathbf{C}(v_i)$:*

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

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

Catmul-Rom Spline Interpolation *of the points $\mathbf{S}^*_{i,j}, i = 1,\dots,n$ by curve $\mathbf{S}^*_j(v)$*

Linearization of the splines *by polygonal lines consisting of $r_\mathbf{C}$ edges*