**Procedural Modeling**

- Subdivision Surfaces.
- Sweeps.
- Fractals.
- Grammar-based modeling.
  - L-systems.
  - Cellular Automata.
  - Lattice Boltzman Machine.

**L-systems**

- A powerful procedural technique
- Based on grammars
  - Rewriting rules
- Most popular for plants
- Recent applications:
  - Cities
  - Feathers

**Plants**

- Complex systems
- Often, have a well defined structure
  - Trunk
  - Big branches
  - Little branches
  - Leaves
- High degree of “recursiveness”
  - Grammars/compilers are good with this.

**Grammar-Based Models**

- Generate description of geometric model by applying production rules

  - $S \rightarrow AB$
  - $A \rightarrow Ba \mid a$
  - $B \rightarrow Ab \mid b$

  $(ab, bab, baab, abaab)$

**Grammar-based models**

- Useful for modeling plants.
  - $R$: turn right
  - $L$: turn left.
Grammar-based models

• Apply the rule randomly to occurrences of $F$. $F \rightarrow F[R]F[L]F^*$. 

Grammar-based model: L-systems

• grammar-based fractal-like models
• describe an object by a string of symbols and provide a set of production rules
• incorporate notions such as branching, pruning, …
• can also vary objects by randomly applying rules
• demo:
  

More examples

Procedural Modeling: summary

• Procedural techniques are very powerful
• Use with care
  – Physical validation is rare
• For some objects, procedure is the answer
  – Plants
• Can complement physics-based methods
  – Adding high frequency details
• General recommendation: add noise to your models to make them more “natural”
Cellular automata

- Simple example - game of life
- The rules:
  - Binary state on a 2D grid: cell / no cell
  - If too few or too many cells surround a cell, it dies
  - If two cells surround empty space, a cell is born
- Very simple rules produce complex behavior
  - Stable patterns
  - Moving stable patterns
  - Oscillations

Complex cellular automata

- Can have more than one cell system interacting
- Rules can be arbitrary
  - Model the needed behavior
- The result is (usually) a distribution of some property
  - Binary result, need extra step
- The art is in creating the rules
  - Take from differential equations
  - Take from general intuition
    - Too many cell -> overcrowding -> dies

Example: clouds

- Need cloud density in 3D
- 3 binary variables
  - Cloud
  - Humidity
  - Action
- General idea:
  - Action tells to create cloud
  - Cloud particle formed if there is humidity
  - Clouds grow, so set action to 1 if there is action nearby

Cloud forming rules

- Hum = (hum) & (!act)
  - Action removes humidity
- Cld = (cld) | (act)
  - Form cloud particle if action
- Act = (!act) & (hum) & f(surround action)
- f= | of all act in a template
- Extinction: do with some probabilities
  - Clear cld
  - Introduce new hum
  - Start new action

Cloud results (Dobashi et. al.)

Lattice Boltzmann Machine

- Extension of cellular automata.
- Can simulate more natural phenomena besides clouds.
**Modeling—summary**

How do we ...
- Represent 3D objects in a computer?
- Construct such representations quickly and/or automatically with a computer?
- Manipulate 3D objects with a computer?

---

**Model Construction**

- Interactive modeling tools
  - CAD programs
  - Subdivision surface editors :)  
- Scanning tools
  - CT, MRI, laser, magnetic, robotic arm, etc.
- Computer vision
  - Stereo, motion, etc.
- Procedural generation
  - Sweeps, fractals, grammars
- Physics-based modeling