Definitions

- **Modeling**
  - Creating a model of an object, usually out of a collection of simpler primitives

- **Primitives**
  - A basic shape handled directly the rendering system
Boundary Representation (BRep)

- Instead of representing a cube as 6 polygons, represent as 6 planes
  - Surfaces are represented as a hollow shell
  - Just the boundary
Common Primitives

- Triangles & Polygons
  - Most common, usually the only choice for interactive
- Patches, Spheres, Cylinders, ...
  - Found in RenderMan
  - Often converted to simpler primitives within the renderer
- Volumes
  - What’s at each point in space?
  - Often with some transparent material
  - Few renderers handle both volume & surface models
Composing Primitives

- Collections of large numbers of primitives
  - Sometimes called Boundary Representation (BRep)

- Constructive Solid Geometry (CSG)
  - Set operations (union, intersection, difference)

- Implicit Models & Blobs
  - Surface where $f(x,y,z) = 0$
  - Sum, product, etc. of simpler functions
Modeling Approaches

- Manual primitive creation
- Procedural
- Scan from physical object
- From data (visualization)
- Through image capture (IBR)
Manual Primitive Creation

- Text Editor
- High-level primitives
- Modeling Programs
Display "cube.tif" "file" "rgba"
Format 1024 768 1
LightSource "distantlight" 1
Projection "perspective"
Translate 0 0 1.5
Rotate 40 -1 1 0
WorldBegin
Surface "matte"
AttributeBegin
Translate -0.5 -0.5 -0.5
Scale 0.02 0.02 0.02
Color [0.02 0.02 0.02]
TransformBegin
Translate 0.5 0.5 0.5
Scale 1 1 1
TransformBegin
Polygon "P" [0.5 0.5 0.5 -0.5 0.5 0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5 ]
Rotate 90 0 1 0
Polygon "P" [0.5 0.5 0.5 -0.5 0.5 0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5 ]
Rotate 90 0 1 0
Polygon "P" [0.5 0.5 0.5 -0.5 0.5 0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5 ]
Rotate 90 0 1 0
Polygon "P" [0.5 0.5 0.5 -0.5 0.5 0.5 -0.5 0.5 -0.5 0.5 -0.5 0.5 ]
Many libraries provide higher level primitives (built atop lower level primitives)
Modeling Programs

- Maya, Blender, LightWave, etc...
Constructive Solid Geometry (CSG)

- Objects are solids
- Boolean operations to combine objects
  - Union
  - Intersection
  - Difference
Constructive Solid Geometry (CSG)
Constructive Solid Geometry (CSG)

- Supported by the RenderMan specification via
  - RiSolidBegin()
  - RiSolidEnd()

- However, this is one of the areas where Pixie's implementation falls short of full specification compliance
  - Currently an unimplemented feature on Pixie's “wish list”
Implicit Functions

- Model as sum of implicit functions
- Surface at threshold
Hybrid Implicit & Polygonal
Procedural

- Describe physical attributes though some (spatial) function
  - Shape
  - Density
  - Color
  - Texture
Procedural Approaches

- Fractals
- Implicit Functions
- Grammars
- Simulations
Fractals

- Complex structure through self-similarity across scales
  - Iterated equations
  - Iterated replacement
  - Spectral synthesis
Fractals – Iterated Equations

- Mandelbrot Set
The Mandelbrot set is defined by a family of complex quadratic polynomials given by $f_c : \mathbb{C} \rightarrow \mathbb{C}$ where $c$ is a complex number.

For each $c$ the following behavior is considered:

$$f_c(z) = z^2 + c$$

Obtained by iterating (starting at $z = 0$)
Thus

\[ f^n_c(z) \]

Denotes the \( n^{th} \) iteration of

\[ f_c(z) \]

The Mandelbrot set is the subset which is constrained as

\[ M = \left\{ c \in \mathbb{C} : \sup_{n \in \mathbb{N}} |f^n_c(0)| < \infty \right\}. \]
Fractals – Iterated Equations

- Mathematical depiction of the Mandelbrot set
  - Point $c$ is black if it belongs to the set
Colored pictures are usually generated by assigning a color to each point which represents how quickly the sequence diverges to infinity $|f^n_c(0)|$.
Fractals – Self Similarity
Fractals – Self Similarity
Fractals – Self Similarity
Fractals – Self Similarity
Fractals – Self Similarity
Fractals – Self Similarity
Fractals – Self Similarity
Fractals – Iterated Replacement

- Koch Curve

Initiator

Generator
Fractals – Iterated Replacement

- Fractal Mountains
Spectral energy a function of frequency
- Higher frequency, less energy
- Characterizes roughness of surface
- Natural phenomena tend to be $1/f$
Fractals – Noise-Based Synthesis

- Band-limited Perlin noise function
  - Most energy between 1/2 and 1 cycle per unit
  - Average value is 0
  - Random, but repeatable
  - 1D, 2D, 3D & 4D versions common

- Sum noise octaves
  - $n(x) + \frac{1}{2}n(2x) + \frac{1}{4}n(4x) + \ldots$
  - Stop add “...” when frequency is too high to see
Perlin Noise
Fractals – Fractal Landscapes

- Landscape height is a fractal function of $x, y$
Fractals – Fractal Landscapes
Fractals – Fractal Landscapes

- Embellishments – textures, shadows, reflections
Further embellishments – atmospheric haze, clouds, waves, colored light source
Multi-Fractal Landscapes

- Change roughness across fractal
  - Scaling (1/2, 1/4, ...) becomes a function
- Here, scale is a function of altitude
Hypertexture

- Add noise or turbulence to functions
Grammar Based Modeling

- Use (mostly) context-free grammars (CFG) to specify structural change over generations
- Often used to simulate a biological growth process
  - Plants
  - Seashells
- L-Systems (Lindenmayer)
A CFG $G = (V, T, S, P)$ where
- $V$ is a set of non-terminals
- $T$ is a set of terminals
- $S$ is the start symbol
- $P$ is a set of production (rules) of the form

$$A \rightarrow x, \text{ where } A \in V, x \in (V \cup T)^*$$
Applying Grammar Rules

- **Symbols**
  - A, B, straight line segments
  - [ ], branch left 90°

- **Rules**
  - $A \rightarrow AA$

- **Strings**
  - $B$
Applying Grammar Rules

- **Symbols**
  - A, B, straight line segments
  - [ ], branch left 45°
  - ( ), branch right 45°

- **Rules**
  - B → A[B]AA(B)
  - A → AA

- **Strings**
  - B
  - A[B]AA(B)
L-System Examples

- **Symbols**
  - \[/\] = push/pop
  - +/- = rotate left/right
  - A-Z = straight segment

- **Rules**
  - 25.7°, Start=X
  - X → F[+X][-X]FX
  - F → FF
L-Systems Example

- **Symbols**
  - Same as previous

- **Rules**
  - 22.5°, Start=X
  - X → F-[[X]+X]+F[+FX]-X
  - F → FF
L-Systems Example

- **Symbols**
  - Same as previous

- **Rules**
  - $22.5^\circ$, Start=$F$
  - $F \rightarrow FF-[-F+F+F]+[+F-F-F]$
L-Systems Additions

- Add any/all of the following
  - 3D Structure
  - Randomness
  - Leaves
  - Flowers
  - Color
L-Systems

- Randomness of 50% introduced in both line segment length and degree of rotation
L-Systems Weeds
L-Systems Plants

Total Fruit number = 36
Total Fruit allocation = 1631
Total vegetative allocation = 1908
Fruit % of total = 46%
L-Systems Plants
L-Systems Flowers
Pruning
Pruning
Simulations

- Biological
  - Simulate growth, development

- Physical
  - Simulate formation or erosion
Biological Simulations
Biological Simulations
Biological Simulations
Biological Simulations
Biological Simulations
Physical Simulations

- Erosion
Scan From Objects

- General concept
  - Find points on surface
  - Connect with mesh
- Mechanical
- Triangulation
  - Laser
  - Structured Light
  - Multiple Cameras
- CAT/MRI scan
Mechanical

- Touch tip to surface
- Measure angles
Triangulation

- Point in space at intersection of ray from $A$ and ray from $B$
Structured Light

- Point in space at intersection of color edge from light source/projector and ray through camera pixel
Visualization

- **Data**
  - Measurements
  - Simulation
  - Information

- **Present visually**
  - Increase understanding
  - Recognize patterns
Visualization

- Can be 3D object
Visualization
Visualization

- Can be 3D, but showing non-visual aspects
Visualization

- Can be not traditionally geometric at all
Image-Based Rendering

- Pixels in one or more cameras
  - Color of point in space
  - Color of light along one ray
- IBR
  - Construct new novel view using only image data
Image Based Rendering