Modeling

CMSC 435/634

Modeling?

Modeling

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

Primitive

A basic shape handled directly the rendering system

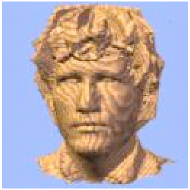
Primitives

Some common primitives

- Triangles & Polygons
  - Most common, usually the only choice for interactive
- Patches, Spheres, Cylinders, ...
  - 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

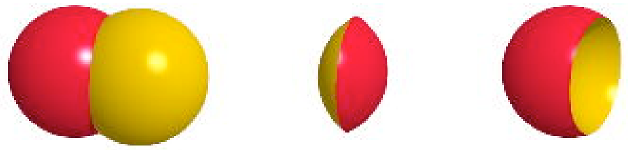
Composing primitives

- 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



Composing primitives

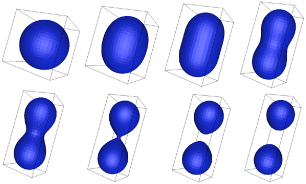
- Collections of large numbers of primitives
  - Sometimes called Boundary Representation (*BRep*)
- Constructive Solid Geometry (*CSG*)
  - Set operations (union, intersection, difference)
- Implicit Models & Blobs
  - Surface where f(x,y,z)=0
  - Sum, product, etc. of simpler functions



Images: Friedrich Lohmueller

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



Images: Paul Bourke

Modeling Approaches

- Manual primitive creation
- Procedural
- Scan from physical object
- From data (visualization)
- Through image capture (image-based rendering)

Modeling Approaches

- Manual primitive creation
- Procedural
  - Fractals
  - Implicit Functions
  - Grammars
  - Simulations
- Scan from physical object
- From data (visualization)
- Through image capture (image-based rendering)

Manual Creation

- Text editor
  - Only very simple primitives and scenes
- High-level primitives
  - Still need to combine several somehow
- Modeling programs
  - Maya, 3D Studio, Houdini, Autocad, Blender, ...

Modeling Approaches

- Manual primitive creation
- Procedural
  - Fractals
  - Implicit Functions
  - Grammars
  - Simulations
- Scan from physical object
- From data (visualization)
- Through image capture (image-based rendering)

Procedural Modeling

- Describe physical attributes through code
  - Shape
    - Output primitives
  - Density
    - Voxels
    - Couple with a conversion or rendering algorithm
  - Color, Texture
    - Enhance an existing shape

Procedural Approaches

- Fractals
- Implicit Functions
- Grammars
- Simulations

Fractals

Complex structure through self-similarity across scales

- Recursive structure
- Small features look *similar* to larger features

Iterated Equations / Mandelbrot Set

$$p' = p^2 + c$$

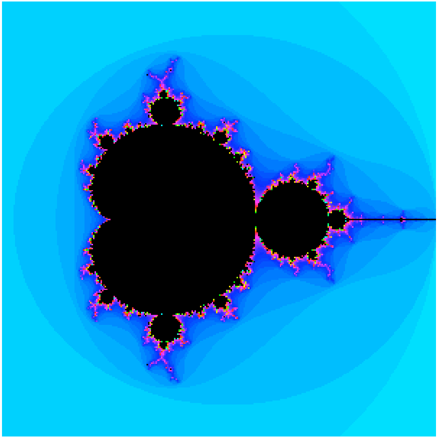
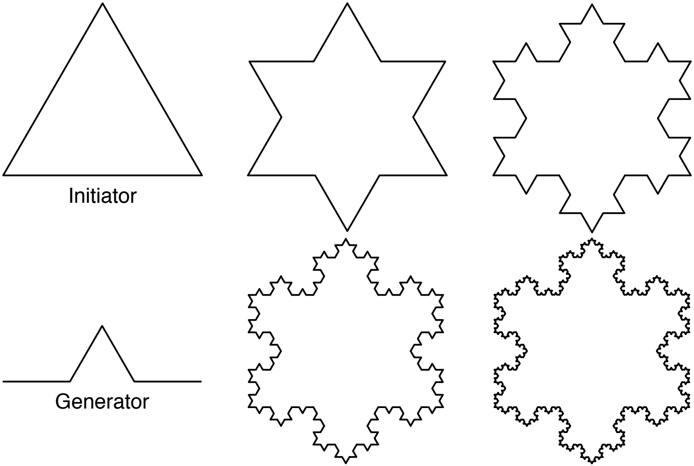


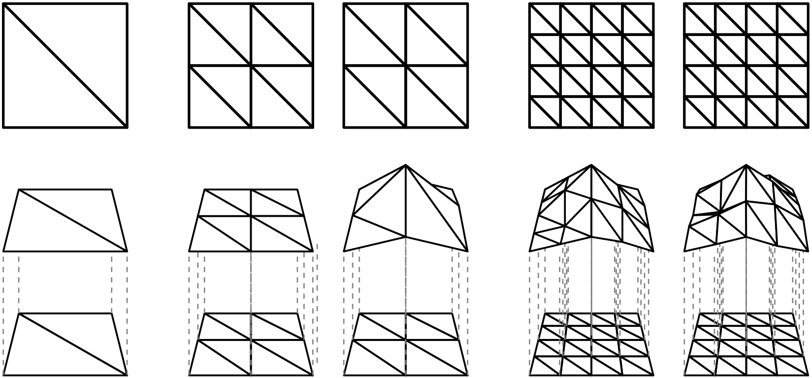
Image: David E. Joyce

Iterated Replacement / Koch Curve



Iterated Replacement / Mountains

Randomness in replacement



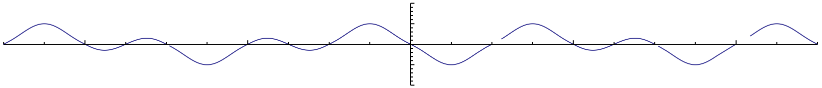
Spectral Synthesis

- Spectral energy a function of frequency
  - Higher frequency, less energy
  - Characterizes roughness of surface
  - Natural phenomena tend to be $1/f$

Noise-Based Synthesis

Band-limited *Perlin noise* function

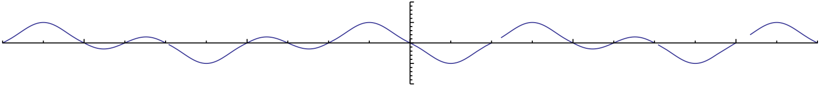
- Most energy between 1/2 and 1 cycle per unit
- Average value is 0
- Random, but repeatable
- 1D, 2D, 3D & 4D versions common



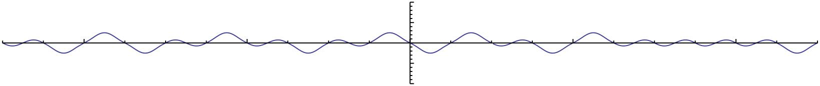
Spectral Synthesis

Sum noise *octaves*

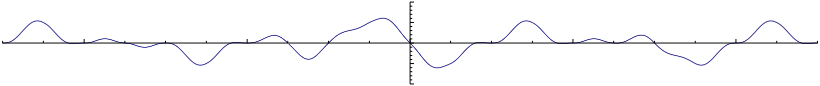
- $n(x) + \frac{1}{2}\ n(2\ x) + \frac{1}{4}\ n(4\ x) + \ldots$
- Stop adding "…" when frequency is too high to see
- Also called *fractional Brownian motion* or *fBm*



$+$



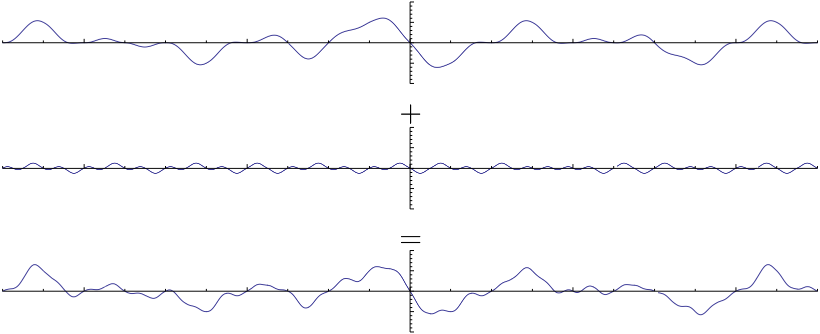
$=$



Spectral Synthesis

Sum noise *octaves*

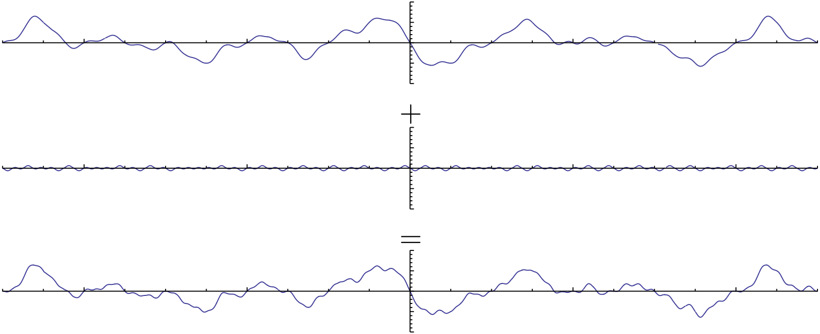
- $n(x) + \frac{1}{2}\ n(2\ x) + \frac{1}{4}\ n(4\ x) + \ldots$
- Stop adding "…" when frequency is too high to see
- Also called *fractional Brownian motion* or *fBm*



Spectral Synthesis

Sum noise *octaves*

- $n(x) + \frac{1}{2}\ n(2\ x) + \frac{1}{4}\ n(4\ x) + \ldots$
- Stop adding "..." when frequency is too high to see
- Also called *fractional Brownian motion* or *fBm*



Fractal Landscape

Landscape height is a fractal function of x,y

- Plus whatever embellishments make it look good

Image: Ken Musgrave

Multifractal

- Change roughness across fractal
  - Scaling $(\frac{1}{2}, \frac{1}{4}, \ldots)$ becomes a function
- Here, scale is a function of altitude

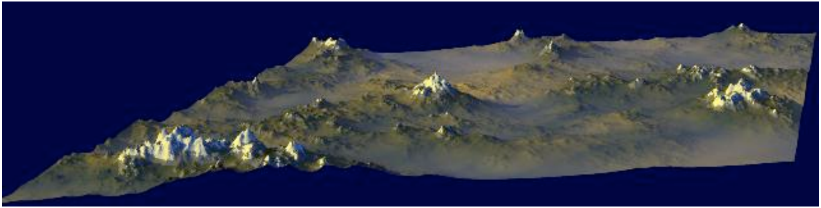
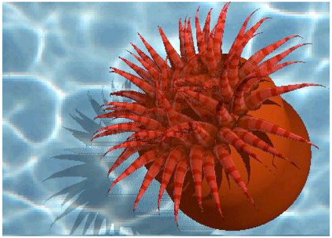


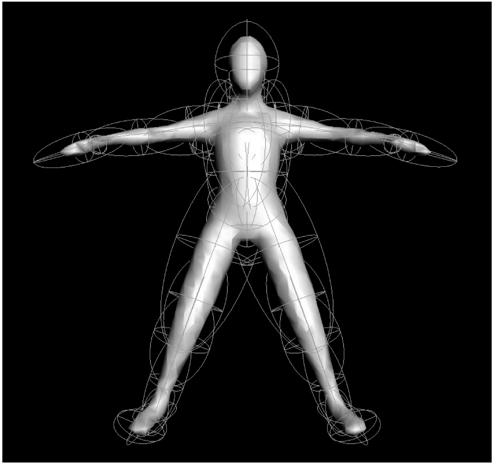
Image: Ken Musgrave

Implicit Functions or Blobby Modeling

- Model as sum of implicit functions
- Surface at threshold



Liang, et al., PG'01

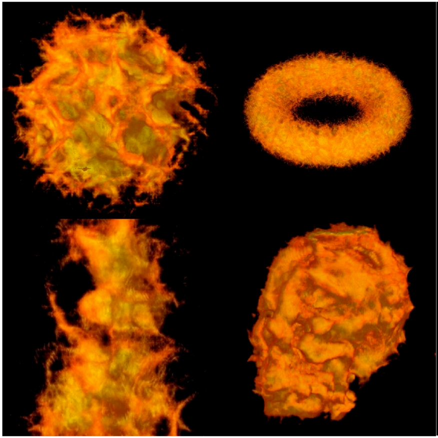
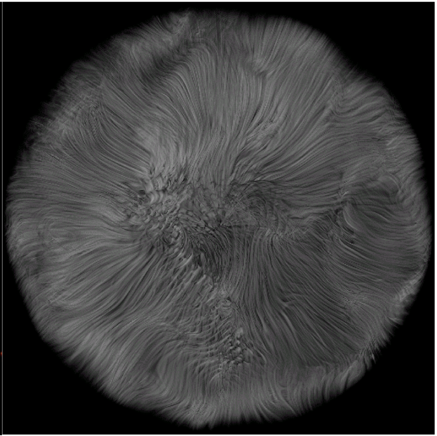


Hybrid Implicit & Polygonal

Bloomenthal, SIGGRAPH 85

Hypertexture

Add noise or turbulence to functions

Perlin & Hoffert, SIGGRAPH 89

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 (Lindenmeyer)

Context-Free Grammar

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 productions (rules) of the form:
  - $A \to x$, where $A \in V, x \in (V \cup T)^*$

Applying Grammar Rules

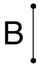
- Symbols
  - A, B, straight line segments
  - [ ], branch left 45°
  - ( ), branch right 45°
- Rules
  - $A \to AA$
  - $B \to A[B]AA(B)$
- Strings
  - B
  - $A[B]AA(B)$
    $AA[A[B]AA(B)]AAAA(A[B]AA(B))$

Applying Grammar Rules

- Symbols
  - A, B, straight line segments
  - [ ], branch left 45°
  - ( ), branch right 45°
- Rules
  - $A \rightarrow AA$
  - $B \rightarrow A[B]AA(B)$
- Strings
  - B
  - $A[B]AA(B)$
  - $AA[A[B]AA(B)]AAAA(A[B]AA(B))$

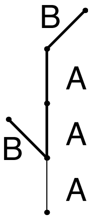
Applying Grammar Rules

- Symbols
  - A, B, straight line segments
  - [ ], branch left 45°
  - ( ), branch right 45°
- Rules
  - $A \rightarrow AA$
  - $B \rightarrow A[B]AA(B)$
- Strings
  - B
  - $A[B]AA(B)$
  - $AA[A[B]AA(B)]AAAA(A[B]AA(B))$



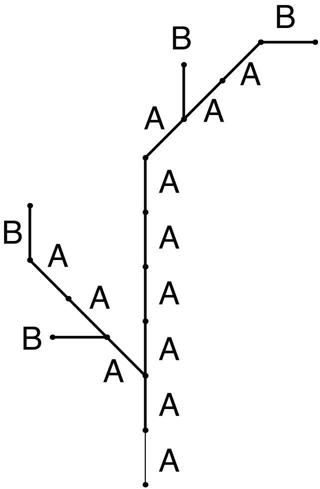
Applying Grammar Rules

- Symbols
  - A, B, straight line segments
  - [ ], branch left 45°
  - ( ), branch right 45°
- Rules
  - $A \rightarrow AA$
  - $B \rightarrow A[B]AA(B)$
- Strings
  - B
  - $A[B]AA(B)$
  - $AA[A[B]AA(B)]AAAA(A[B]AA(B))$



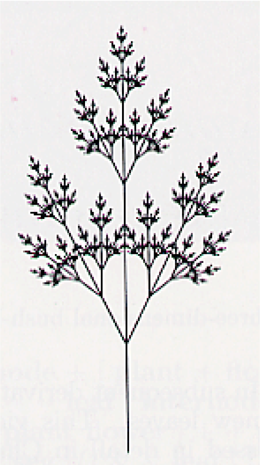
Applying Grammar Rules

- Symbols
  - A, B, straight line segments
  - [ ], branch left 45°
  - ( ), branch right 45°
- Rules
  - $A \rightarrow AA$
  - $B \rightarrow A[B]AA(B)$
- Strings
  - B
  - $A[B]AA(B)$
  - $AA[A[B]AA(B)]AAAA(A[B]AA(B))$



L-System Examples

- Symbols
  - $[/]$ = push/pop
  - $+/-$ = rotate left/right
  - $A - Z$ = straight segment
- Rules
  - $25.7°$, 7 generations
  - $X \rightarrow F[+X][-X]FX$
  - $F \rightarrow FF$



L-System Examples

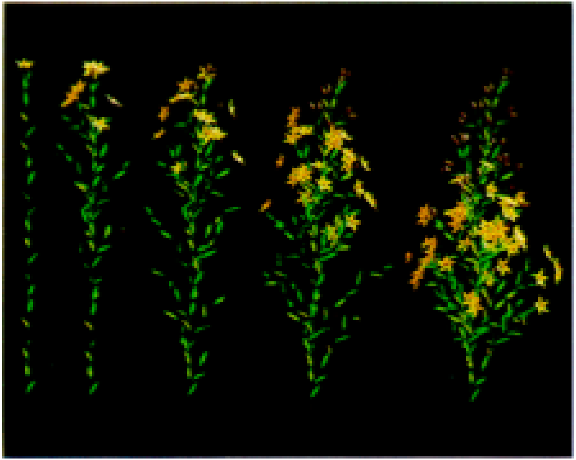
- Rules
  - 22.5°, 5 generations
  - $X \rightarrow F{-}[[X]{+}X]{+}F[{+}FX]{-}X$
  - $F \rightarrow FF$

L-System Examples

- Rules
  - 22.5°, 4 generations
  - $F \to FF - [F + F + F] + [+F - F - F]$

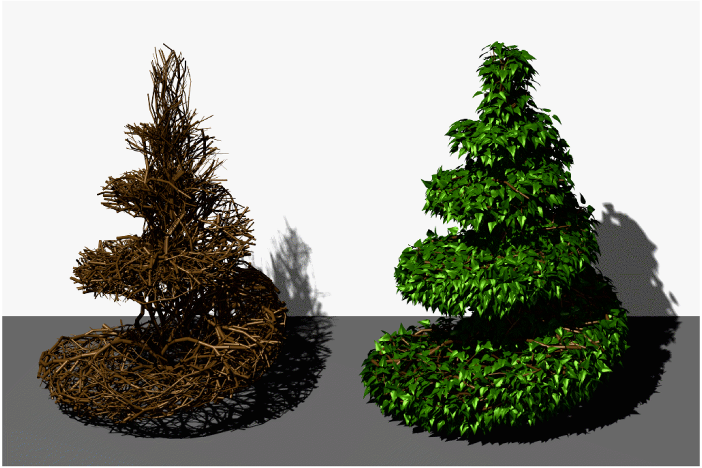
Additions

- 3D structure
- Randomness
- Leaves
- Flowers



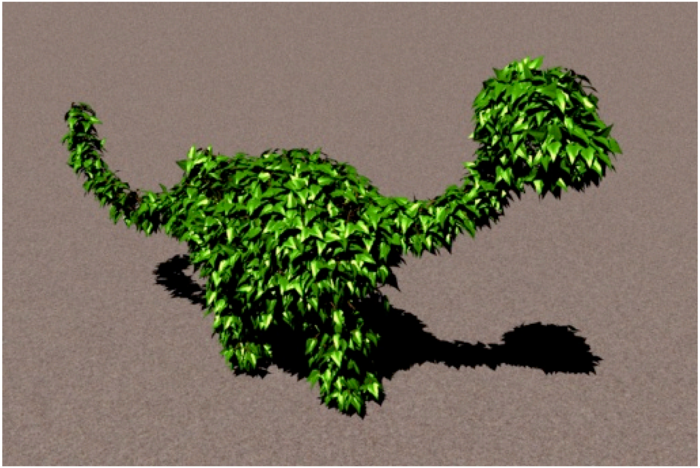
Prusinkiewicz, et al., SIGGRAPH 88

Pruning



Prusinkiewicz, et al., SIGGRAPH 94

Pruning



Prusinkiewicz, et al., SIGGRAPH 94

Simulations

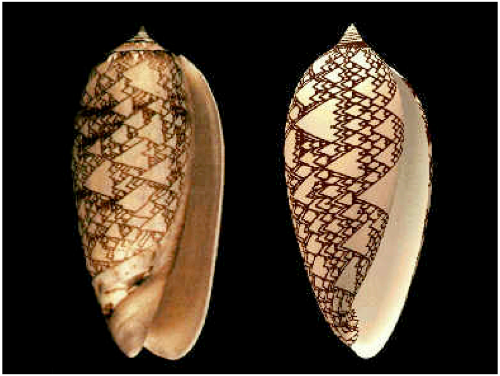
Biological

- Simulate growth, development

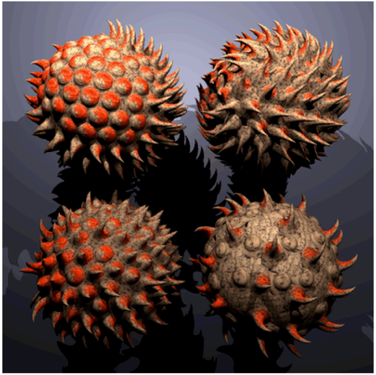
Physical

- Simulate formation or erosion

Biological Simulations



Fowler, et al., SIGGRAPH 92

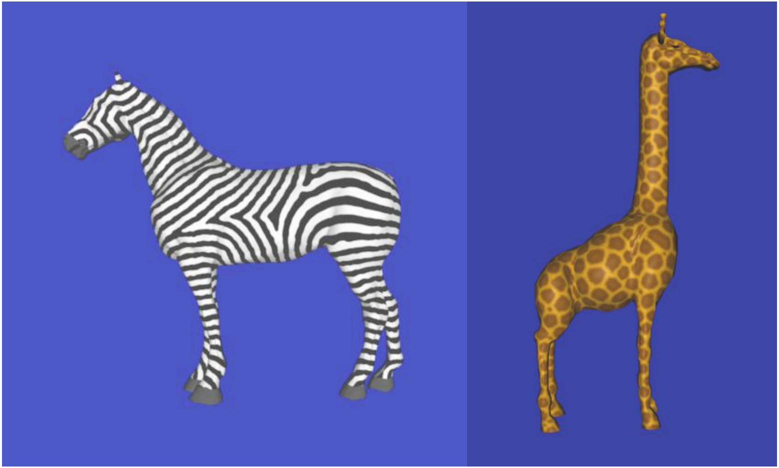


Fleischer, et al., SIGGRAPH 95

Biological Simulations

Fowler, et al., SIGGRAPH 92

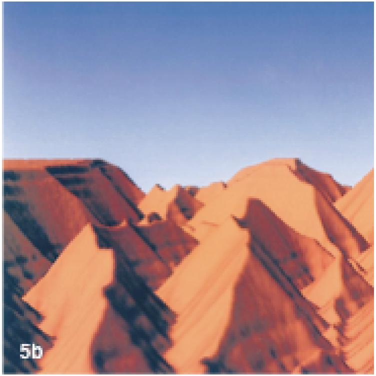
Biological Simulations



Turk, SIGGRAPH 91

Physical Simulation

Erosion, Deposition



Kenji Nagashima, Visual Computer 1997

Modeling Approaches

- Manual primitive creation
- Procedural
  - Fractals
  - Implicit Functions
  - Grammars
  - Simulations
- Scan from physical object
- From data (visualization)
- Through image capture (image-based rendering)

Scan from Objects

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

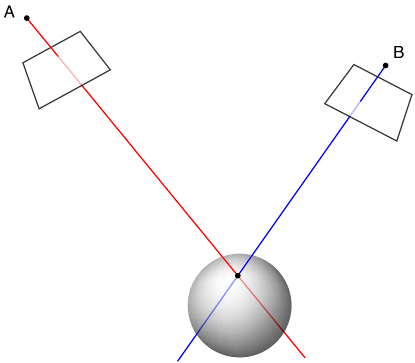
Mechanical

- Touch tip to surface
- Measure angles

Triangulation

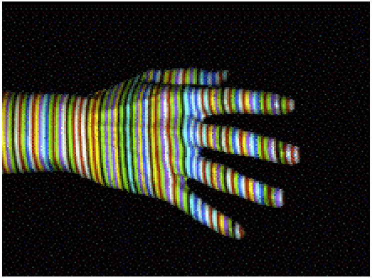
Point in space at intersection

- Ray from light A
- Ray through pixel B

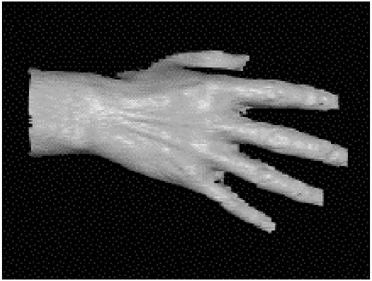


Structured Light

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



projected pattern



resulting model

Zhang, Curless and Seitz, 3DPVT 2002

Modeling Approaches

Manual primitive creation

Procedural

- Fractals
- Implicit Functions
- Grammars
- Simulations

Scan from physical object

From data (visualization)

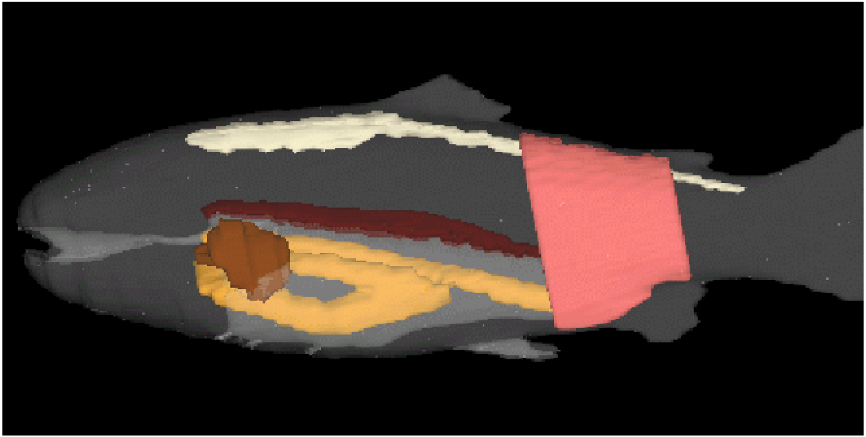
Through image capture (image-based rendering)

Visualization

- Data
  - measurements
  - simulation
  - information
- Present visually
  - Increase understanding
  - Recognize patterns

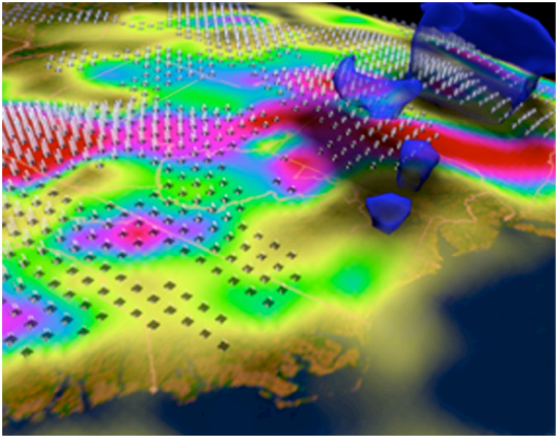
Visualization

Can be 3D Object



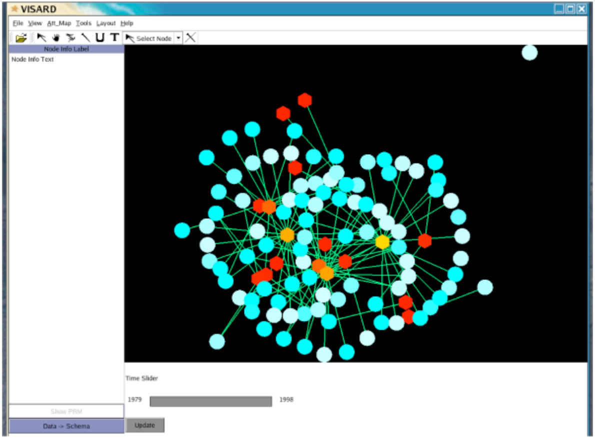
Visualization

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



Visualization

Can be not traditionally geometric at all



Modeling Approaches

- Manual primitive creation
- Procedural
  - Fractals
  - Implicit Functions
  - Grammars
  - Simulations
- Scan from physical object
- From data (visualization)
- Through image capture (image-based rendering)

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