MPRI 2.39 - General Modeling Approaches
General Modeling Approaches

- Manual/Interactive Modeling
- Procedural Modeling
- Surface Acquisition
General Modeling Approaches

- Manual/Interactive Modeling
  - Polygonal Mesh Modeling
    - Parametric Surface Modeling (viewed in the first class)
    - Implicit Surface Modeling (detailed later)
  - Procedural Modeling
- Surface Acquisition
Polygonal Mesh Modeling

- The most commonly used for 3D production
- Basic of 3D modeling softwares
  *Maya, 3DStudio, Cinema4D, Blender, etc.*

**General Principle**

- Define a low resolution mesh:
  - Start from basic primitive
  - Extends it by extrusion
  - Iteratively refines it to add details
  - Possibly smooth it using subdivision surfaces

+ Fully interactive process. Total freedom.
+ Good mesh quality (quads).
  *allows deformation, animation*
- Requires artists skills. Tedious.
- Iterative refinement
  - Low resolution must first be designed to handle details afterward
  - No future merging/splittings
General Modeling Approaches

- Manual/Interactive Modeling
- **Procedural Modeling**
- Surface Acquisition
Procedural modeling

**Objective** Modeling huge, detailed geometry (too tedious for artists)

Applied to complex shapes or scenes

*ex.: Natural scenes, planets, cities, etc*

**Main approaches**

- Noise and fractals
- Rules/grammar based
- Simulation

+ Powerful approach to generate complex shape
- Only indirect control over the shape, may be hard to tune
Fractals

**Idea:** Recursively add self-similar details
Simple rule $\Rightarrow$ complex shapes
May look like complex natural details

Koch Snowflake

Sierpinski triangle, Shell: Oliva Porphyria
Perlin Noise

A widely used noise function.

*Original article* [An Image Synthesizer, Ken Perlin, SIGGRAPH 85]

Continuous Pseudo-random function
- Spatial variations are continuous, but non periodic
- Function can be evaluated at any point - deterministic

**Creating a smooth function**
- Compute pseudo-random gradient at each sample point
  
  *Use some hash function*

- Interpolate smooth cubic curve between each points

(Algorithm for nD: Simplex noise)
Fractal Perlin Noise

Sum over multiple instance with increasing frequencies

\[ f : \text{Smooth Perlin Noise function} \]

\[ g(x) = \sum_{k=0}^{N} \alpha^k f(\omega^k x) \]

- \( N \) number of Octave
- \( \alpha \) attenuation (1/\( \alpha \) persistence)
- \( \omega \) frequency gain
Perlin noise usage

Material texture
- Ridge effect: $\sum_k \alpha^k |f(\omega^k x)|$
- Marble effect: $\sin(x + \sum_k \alpha^k |f(\omega^k x)|)$

Animated textures
- Translation: $f(x, y + t)$
- Smooth evolution: $f(x, y, t)$

Mountain-looking terrain
- $z = f(x, y)$
Perlin noise applications

In almost every complex shapes ...

Look at Shader Toy + Noise (example)
Perlin Noise Terrain

Consider the surface $S$ defined as

$$S(u, v) = \begin{cases} 
   x(u, v) = u \\
   y(u, v) = v \\
   z(u, v) = h \cdot g(s(u + o), s(v + o)) 
\end{cases}$$

The perlin noise

$$g(u, v) = \sum_{k=0}^{N} \alpha^k f(2^k u, 2^k v)$$

$$N = 9 \\
\alpha = 0.4 \\
h = 0.3 \\
s = 1 \\
o = 0$$

Which parameters correspond to (a,b,c,d,e,f) terrains?
Rules based: L-systems

- L-systems: Lindenmayer Aristid, Biologist.
- Developed later on graphics by Przemyslaw Prusinkiewicz
Rules based

Generalization to **shape grammars**

Define

- **Symbols** - Set of basic shapes
- **Rules** - Replacement functions
  
  *Stochastic variations, applying with probabilities, etc.*
- **Axiom** - Initial shape

=> **Derivation** until terminal symbols

ex. Modeling cities

Lot -> extrude(10) Mass
Mass -> FaceSplit { sides: Facade }
Facade -> Split("y") { 3: FirstFloor, ~1: TopFloors }
TopFloors -> Repeat("y"){ 1 : Floor }
Floor -> Repeat("x"){ 1 : Window }
Window -> insert("window.obj")
Physics inspired

Use physics equation to generate shapes.

- Old principle (genetic algorithms, etc)
  ex. [William Reeves, Particle Systems - A technique for Modeling a class of Fuzzy Objects. ACM TOG 83]

- Came back recently for shape optimization in 3D additive printing
  
  *Strong structure, low material consumption*
General Modeling Approaches

- Manual/Interactive Modeling
- Procedural Modeling
- Surface Acquisition
Acquisition

**Objective** - Acquire real-life 3D shape

**Methodology**
1. Calibration of acquisition system
2. Data acquisition
3. Reconstruction to create surfaces

**Technologies**
- Optical: cheap, fast, low resolution
- Laser: expensive, slow, high resolution.

**Main acquisition system**
- Multi-views images
- Depth camera (structured lights)
- Laser scanner

+ Intrinsically represents **realistic** data
- No control over the acquired shape
- Shapes are hard to reuse: bad triangle quality, high number of samples
- Object to be modeled must available
Acquisition system

**Multi-view images**
- Several images of the same object from different view-points
- Compute depth of corresponding points

**Structured lighting**
- Project image pattern (IR spectrum)
- Compute depth value from pattern deformation
  *ex. Kinect V1*

**Time of flight**
- Acquire time of reflection for short light pulses
- Distance as function of time
  *ex. Kinect V2, LIDAR*
Surface Reconstruction

**Input** Set of scattered 3D positions in space (no normals)

**Delaunay triangulation**
- 3D Crust algorithm [Amenta et al., SIGGRAPH 98]
- PowerCrust [Amenta et al., SMA, 01]
- Cocone [Dey and Goswami, JCISE, 03]

  + Accurate to data - all points are used
  - Sensible to noise, outliers

**Implicit surface reconstruction**
- Distance field to tangent plane [Hoppe et al. SIGGRAPH 92]
- Radial Basis Functions (RBF) [Carr et al., SIGGRAPH 01]
- Poisson surface reconstruction [Kazhdan et al., SGP 2006]

  + Handle noisy data
  - May depends on sampling
  - Smooth-out details

State of the Art:
[A Survey of Surface Reconstruction from Point Clouds, Berget et al., EG STAR, 2016]
MPRI 2.39 - Modeling with Implicit Surfaces
Limits of boundary representations modeling

B-Rep

Meshes, Parametric surfaces, Subdivision surfaces, etc.
- Topology modification hard to integrate
- Boundary surface doesn't necessarily represents a volume
  ex. Klein bottles, self-intersections
See Combinatorial Maps for robust encoding of Brep representation.

Implicit surfaces

\[ S = \{(x, y, z) \in \mathbb{R}^3 \mid f(x, y, z) = iso\} \]

Intrinsically based on volumetric representation
- Robust volume modeling
- Easy blending/merging between pieces
Properties of implicit surfaces

\[ S = \{ p \in \mathbb{R}^3 \mid f(p) = iso \} \]

- Position \( p \) on the surface: \( f(p) = iso \)
- Position \( p \)
  - inside the surface \( f(p) > iso \)
  - outside the surface \( f(p) \)
  \( \Rightarrow \) Easy query for interior/exterior position
- Normal on the surface \( n = \nabla f(p) \)

Consider \( dp \) infinitesimal vector within the tangent plane of the surface at point \( p \)

\[
f(p) = f(p + dp) = iso \\
\Rightarrow f(p) - f(p + dp) = 0 \\
\Rightarrow dp \cdot \nabla f(p) = 0
\]
Encoding scalar field

Discrete representation
Grid based: one value per voxel
*Naturally obtained from scanner acquisition* (High memory cost)

Continuous representation
From optimization (ex. Radial Basis Functions)
User defined (ex. Skeleton based)
Blobs

General idea (1982)

- Decreasing function $f_i$ of distance to a point $p_i$
- $p_i$ is called a skeleton

- Exemple **Exponential field**

$$f_i(p) = K_i \exp \left( - \frac{||p - p_i||^2}{R_i^2} \right)$$

_Blinn 1982_

- Consider $K_0 = 1, R_0 = 1, p_0 = (0.0, 0.0, 0.0)$, what is $f_0(p) = 0.5$ ?
- Consider $K_1 = 1, R_1 = 1, p_1 = (1.0, 0.0, 0.0)$, what is
  $$\max(f_0(p), f_1(p)) = 0.5$$
  $$\min(f_0(p), f_1(p)) = 0.5$$
- Consider $K_1 = 1, R_1 = 1, p_1 = (2.0, 0.0, 0.0)$, what is
  $$f_0(p) + f_2(p) = 0.5$$
Blobs max/min

$f_0$  

$f_1$  

$max(f_0, f_1)$  

$min(f_0, f_1)$
Blobs sum

\[ f_0 + f_2 \]
Function blending

\[ f_i(p) = K_i \exp\left(-\frac{||p - p_i||^2}{R_i^2}\right) \]

Moving two skeletons toward each other \( f(p) = \sum_i f_i(p) \).

- What is the role of \( K_i \) and \( R_i \)?
- What happens when \( K_i \) is negative?
- Model benefits/drawbacks: smoothness? local control?
Compact support

Field function with compact support $\Rightarrow$ local control of implicit surface

Distance function $f(d), d > 0$

Metaballs

Nishimura 1985

$$f(d) = \begin{cases} 
1 - 3d^2 & 0 < d \leq 1/3 \\
3/2 (1 - d)^2 & 1/3 < d \leq 1 \\
0 & d > 1 
\end{cases}$$

$C^1$ function

Soft Objects

Wyvill 1986

$$f(d) = -4/9 d^2 + 17/9 d^4 - 22/9 d^2 + 1, 0 < d < 1$$

$C^2$ function
Consider the implicit surface defined as
\[ f(p) = \exp(-\|p\|^2) \]

What is the implicit surface corresponding to
- \( f(p + (1, 0, 0)) \)
- \( f(2p) \)

More generally
- \( f \circ T(p) \), where \( T \) is a general transformation
Transformation

Given a space deformation $T$ on the implicit surface
The associated field is $f_{deformed} = f \circ T^{-1}$.

Common deformations example

$$T = \begin{pmatrix} s(z) & 0 & 0 \\ 0 & s(z) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T = \begin{pmatrix} \cos(\theta(z)) & \sin(\theta(z)) & 0 \\ -\sin(\theta(z)) & \cos(\theta(z)) & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

$$T = \begin{pmatrix} \cos(\theta(z)) & 0 & -\sin(\theta(z)) \\ 0 & 1 & 0 \\ \sin(\theta(z)) & 0 & \cos(\theta(z)) \end{pmatrix}$$
Segment skeleton

- Naive solution: field function based on (shortest cartesian) distance to the segment.
  
ex. $f(p) = \exp(-d(p, s)^2)$

- Problem: Bulge at junction between two segments.
Convolution surfaces

- Integral of infinitesimal contributions along skeleton

\[ f(p) = \int_{q \in \text{skeleton}} \kappa(\|p - q\|) \, dq \]

\( \kappa \) : kernel

Bloomenthal, Shoemake. Convolution Surfaces. ACM SIGGRAPH 1991

- Sum of fields = Field generated by all skeletons

\[ f(p) = \sum_i f_i(p) = \int_{q \in C} \sum_i \kappa(\|p - q_i\|) \, dq_i \]
Convolution surfaces: Cauchy Kernel

Cauchy Kernel $\kappa(x) = \frac{1}{(1 + \alpha^2 x^2)^2}$

$$f(p) = \int_{q \in \text{skelton}} \frac{1}{(1 + \alpha^2 \|p - q\|^2)^2} \, dq, \quad q = p_0 + s(p_1 - p_0), \quad s \in [0, 1]$$

$$f(p) = \frac{1}{2p^2} \left( \frac{h}{p^2 + \alpha^2 h^2} + \frac{L - h}{\alpha^2 (L - h)^2 + p^2} \right) + \frac{1}{2\alpha p^3} \left( \tan \left( \frac{sh}{p} \right) + \tan \left( \frac{s(L - h)}{p} \right) \right)$$

$L = \|p_1 - p_0\|

d = \|p - p_0\|

h = (p - p_0) \cdot (p_1 - p_0) / L

p = 1 + \alpha^2 (d^2 - h^2)$
Generalization to triangular skeleton

\[ f(p) = \iint_{q \in \text{skeleton}} \kappa(\|p - q\|) \, dq \]

Explicit formula for Cauchy Kernel

J. McCormack, A. Sherstyuk. Creating and Rendering Convolution Surfaces. CGF 98.

\[
F_{\text{triangle}}(r) = \frac{1}{2qs} \left( \frac{n}{A} \left( \frac{s(1 + a_1 + u)}{A} \right) + \frac{m}{B} \left( \frac{s(a_2 - u)}{B} \right) + \frac{v}{C} \left( \frac{s(a_1 + u)}{C} + \frac{s(a_2 - u)}{C} \right) \right)
\]

where

\[
\begin{align*}
A^2 &= a_1^2w + h^2(q + s^2u^2) - 2hs^2a_1ng, \\
B^2 &= a_2^2w + h^2(q + s^2u^2) + 2hs^2a_2ug, \\
C^2 &= 1 + s^2(d^2 - u^2), \\
g &= v - h, \\
q &= 1 + s^2(d^2 - u^2 - v^2), \\
w &= c^2 - 2hs^2v + h^2s^2, \\
m &= a_2g + uh, \\
n &= uh - a_1g
\end{align*}
\]
Blending Operators

- $\max(f_1(p), f_2(p)), \min(f_1(p), f_2(p)), f_1(p) + f_2(p)$ are not the only possible operators

- Ricci operators (Clean Union)
  \[ f(p) = (f_1^n(p) + f_2^n(p))^{1/n} \]

- Pasko operators
  A. Pasko et al. Function Representation in Geometric Modeling. TVC 95
  \[ f(p) = f_1(p) + f_2(p) - \sqrt{f_1^2(p) + f_2^2(p)} + \frac{a_0}{1 + \left(\frac{f_1}{a_1}\right)^2 + \left(\frac{f_2}{a_2}\right)^2} \]
Graphical interpretation of blending operators

Representation in the space $f_1 - f_2$

(a) Sum operator  (b) Ricci-3 operator  (c) Ricci-10 operator  (d) Union operator

images from Valentin Roussellet
Gradient Blending

Taking into account gradients of fields

\[ f = \mathcal{O}(f_1, f_2, \nabla f_1, \nabla f_2) \]

O. Gourmel et al. A Gradient-Based Implicit Blend. TOG
Blob Tree

Modeling a CSG construction tree

- Terminal leaves are skeleton
- Binary/N-ary edges are operators
- Unary edges are space deformations

Erwin de Groot and Brian Wyvill