3D Modeling
COS 426, Spring 2018
Princeton University
Adam Finkelstein
Syllabus

I. Image processing
II. Modeling
III. Rendering
IV. Animation

Image Processing
(Rusty Coleman, CS426, Fall99)

Modeling
(Denis Zorin, CalTech)

Rendering
(Michael Bostock, CS426, Fall99)

Animation
(Angel, Plate 1)
What is 3D Modeling?

• Topics in computer graphics
  • Imaging = representing 2D images
  • Modeling = representing 3D objects
  • Rendering = constructing 2D images from 3D models
  • Animation = simulating changes over time
Blender demo reel 2016 (musimduift)
Modeling

• How do we ...  
  • Represent 3D objects in a computer?  
  • Acquire computer representations of 3D objects?  
  • Manipulate computer representations of 3D objects?
Modeling Background

• Scene is usually approximated by 3D primitives
  • Point
  • Vector
  • Line segment
  • Ray
  • Line
  • Plane
  • Polygon
3D Point

• Specifies a location
  • Represented by three coordinates
  • Infinitely small

```c
typedef struct {
    Coordinate x;
    Coordinate y;
    Coordinate z;
} Point;
```

• $(x,y,z)$

Origin
3D Vector

- Specifies a direction and a magnitude
  - Represented by three coordinates
  - Magnitude $\|V\| = \sqrt{dx \, dx + dy \, dy + dz \, dz}$
  - Has no location

```c
typedef struct {
    Coordinate dx;
    Coordinate dy;
    Coordinate dz;
} Vector;
```

(dx, dy, dz)
3D Vector

• Dot product of two 3D vectors
  • $\mathbf{V}_1 \cdot \mathbf{V}_2 = ||\mathbf{V}_1|| \cdot ||\mathbf{V}_2|| \cdot \cos(\Theta)$
3D Vector

• Cross product of two 3D vectors
  • $V_1 \times V_2 = \text{vector perpendicular to both } V_1 \text{ and } V_2$
  • $||V_1 \times V_2|| = ||V_1|| \cdot ||V_2|| \cdot \sin(\Theta)$

\[
(d_{x_1}, d_{y_1}, d_{z_1})
\]

\[
(d_{x_2}, d_{y_2}, d_{z_2})
\]

$V_1 \times V_2$
3D Line Segment

• Linear path between two points
  • Parametric representation:
    • \( P = P_1 + t (P_2 - P_1), \quad (0 \leq t \leq 1) \)

```c
typedef struct {
    Point P1;
    Point P2;
} Segment;
```
3D Ray

• Line segment with one endpoint at infinity
  • Parametric representation:
    • \( P = P_1 + t \ V, \quad (0 \leq t < \infty) \)

```c
typedef struct {
    Point P1;
    Vector V;
} Ray;
```
3D Line

• Line segment with both endpoints at infinity
  • Parametric representation:
    • \( P = P_1 + t \mathbf{V}, \ (-\infty < t < \infty) \)

```c
typedef struct {
    Point P1;
    Vector V;
} Line;
```
3D Plane

• A linear combination of three points
3D Plane

• A linear combination of three points
  • Implicit representation:
    • \( P \cdot N - d = 0 \), or
    • \( ax + by + cz + d = 0 \)

  \[\text{typedef struct \{ Vector N; Distance d; \} Plane;}\]

  • \( N \) is the plane “normal”
    • Unit-length vector
    • Perpendicular to plane
3D Polygon

• Set of points “inside” a sequence of coplanar points

```
typedef struct {
    Point *points;
    int npoints;
} Polygon;
```

Points are in counter-clockwise order
3D Object Representations

How can this object be represented in a computer?
3D Object Representations

How about this one?
3D Object Representations

This one?

Wallpapersonly.net
3D Object Representations

This one?

Solidworks
3D Object Representations

This one?

The visible human
3D Object Representations

This one?  FumeFx
3D Object Representations

- **Points**
  - Range image
  - Point cloud

- **Surfaces**
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit

- **Solids**
  - Voxels
  - BSP tree
  - CSG
  - Sweep

- **High-level structures**
  - Scene graph
  - Application specific
Equivalence of Representations

• Thesis:
  • Each representation has enough expressive power to model the shape of any geometric object
  • It is possible to perform all geometric operations with any fundamental representation

• Analogous to Turing-equivalence
  • Computers and programming languages are Turing-equivalent, but each has its benefits…
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation

Data structures determine algorithms
Why Different Representations?

Efficiency for different tasks

• Acquisition
  • Range Scanning
• Rendering
• Analysis
• Manipulation
• Animation

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

• Acquisition
  • Computer Vision
• Rendering
• Analysis
• Manipulation
• Animation
Why Different Representations?

Efficiency for different tasks

- Acquisition
  - Tomography
- Rendering
- Analysis
- Manipulation
- Animation

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
  - Intersection
- Analysis
- Manipulation
- Animation
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Curvature, smoothness
- Manipulation
- Animation

Analysis of surface quality

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Fairing
- Manipulation
- Animation

Surface smoothing for noise removal

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Parametrization
- Manipulation
- Animation
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Texture mapping
- Manipulation
- Animation

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

• Acquisition
• Rendering
• Analysis
  • Reduction
• Manipulation
• Animation
Why Different Representations?

 Efficiency for different tasks

• Acquisition
• Rendering
• Analysis
  • Structure
• Manipulation
• Animation

Extracting shape structure

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Symmetry detection
- Manipulation
- Animation
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Correspondence
- Manipulation
- Animation

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Shape retrieval
- Manipulation
- Animation

Shao et al. 2011
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Segmentation
- Manipulation
- Animation

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
  - Composition
- Manipulation
- Animation

Lin et al. 2008
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
  - Deformation
- Animation
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
  - Deformation
- Animation

Freeform and multiresolution modeling

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
  - Control
- Animation
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
  - Healing
- Animation

Removal of topological and geometrical errors

DGP course notes, Technion
Why Different Representations?

Efficiency for different tasks

• Acquisition
• Rendering
• Analysis
• Manipulation
• Animation
  • Rigging
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation
  - Deformation transfer

Sumner et al. 2004
Why Different Representations?

Efficiency for different tasks

- Acquisition
- Rendering
- Analysis
- Manipulation
- Animation
  - Simulation

Physically Based Modelling course notes, USC
Why Different Representations?

Efficiency for different tasks

• Acquisition
• Rendering
• Analysis
• Manipulation
• Animation
  • Fabrication
3D Object Representations

• Points
  • Range image
  • Point cloud

• Surfaces
  • Polygonal mesh
  • Subdivision
  • Parametric
  • Implicit

• Solids
  • Voxels
  • BSP tree
  • CSG
  • Sweep

• High-level structures
  • Scene graph
  • Application specific
3D Object Representations

- Points
  - Range image
  - Point cloud

- Surfaces
  - Polygonal mesh
  - Subdivision
  - Parametric
  - Implicit

- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep

- High-level structures
  - Scene graph
  - Application specific
Range Image

Set of 3D points mapping to pixels of depth image

• Can be acquired from range scanner
Point Cloud

Unstructured set of 3D point samples

- Acquired from range finder, computer vision, etc
3D Object Representations

- Points
  - Range image
  - Point cloud

- Surfaces
  - Polyhedral mesh
  - Subdivision
  - Parametric
  - Implicit

- Solids
  - Voxels
  - BSP tree
  - CSG
  - Sweep

- High-level structures
  - Scene graph
  - Application specific
Polygonal Mesh

Connected set of polygons (often triangles)
Subdivision Surface

Coarse mesh & subdivision rule

- Smooth surface is limit of sequence of refinements

Zorin & Schroeder
SIGGRAPH 99
Course Notes
Parametric Surface

Tensor-product spline patches

- Each patch is parametric function
- Careful constraints to maintain continuity

\[
\begin{align*}
x &= F_x(u,v) \\
y &= F_y(u,v) \\
z &= F_z(u,v)
\end{align*}
\]

FvDFH Figure 11.44
Implicit Surface

Set of all points satisfying: $F(x,y,z) = 0$

Polygonal Model

Implicit Model

Bill Lorensen
SIGGRAPH 99
Course #4 Notes
3D Object Representations

• Points
  • Range image
  • Point cloud

• Surfaces
  • Polygonal mesh
  • Subdivision
  • Parametric
  • Implicit

• Solids
  • Voxels
  • BSP tree
  • CSG
  • Sweep

• High-level structures
  • Scene graph
  • Application specific
Voxel grid

Uniform volumetric grid of samples:

- Occupancy (object vs. empty space)
- Density
- Color
- Other function (speed, temperature, etc.)

- Often acquired via simulation or from CAT, MRI, etc.
Octree

The adaptive version of the voxel grid

- Significantly more space efficient
- Makes operations more cumbersome
BSP Tree

Hierarchical Binary Space Partition with solid/empty cells labeled

- Constructed from polygonal representations
Constructive Solid Geometry: set operations (union, difference, intersection) applied to simple shapes

FvDFH Figure 12.27

H&B Figure 9.9
Sweep

Solid swept by curve along trajectory

Removal Path

Sweep Model

Bill Lorensen
SIGGRAPH 99
Course #4 Notes
3D Object Representations

• Points
  • Range image
  • Point cloud

• Surfaces
  • Polygonal mesh
  • Subdivision
  • Parametric
  • Implicit

• Solids
  • Voxels
  • BSP tree
  • CSG
  • Sweep

• High-level structures
  • Scene graph
  • Application specific
Scene Graph

Union of objects at leaf nodes

Bell Laboratories

avalon.viewpoint.com
Application Specific

Apo A-1
(Theoretical Biophysics Group,
University of Illinois at Urbana-Champaign)

Architectural Floorplan
(CS Building, Princeton University)
Taxonomy of 3D Representations

3D Shape
  - Discrete
    - Voxels, Point sets
  - Continuous
    - Combinatorial
      - Topological
        - Mesh Subdivision
      - Set Membership
        - BSP Tree
        - Cell Complex
    - Functional
      - Parametric
        - Bezier
        - B-Spline
      - Implicit
        - Algebraic
Equivalence of Representations

• Thesis:
  • Each representation has enough expressive power to model the shape of any geometric object
  • It is possible to perform all geometric operations with any fundamental representation

• Analogous to Turing-equivalence
  • Computers and programming languages are Turing-equivalent, but each has its benefits…
Computational Differences

- Efficiency
  - Representational complexity (e.g. surface vs. volume)
  - Computational complexity (e.g. $O(n^2)$ vs $O(n^3)$)
  - Space/time trade-offs (e.g. tree data structures)
  - Numerical accuracy/stability (e.g. degree of polynomial)

- Simplicity
  - Ease of acquisition
  - Hardware acceleration
  - Software creation and maintenance

- Usability
  - Designer interface vs. computational engine
Upcoming Lectures

• Points
  • Range image
  • Point cloud

• Surfaces
  • Polygonal mesh
  • Subdivision
  • Parametric
  • Implicit

• Solids
  • Voxels
  • BSP tree
  • CSG
  • Sweep

• High-level structures
  • Scene graph
  • Application specific