3D Rendering

Thomas Funkhouser
Princeton University
COS 426, Spring 2006

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

Rendering
• Generate an image from geometric primitives

Overview
• 3D scene representation
• 3D viewer representation
• Visible surface determination
• Lighting simulation

Where Are We Now?
I. Image processing
II. Rendering
III. Modeling
IV. Animation

3D Rendering Example
What issues must be addressed by a 3D rendering system?
Overview

- 3D scene representation
- 3D viewer representation
- Visible surface determination
- Lighting simulation

How is the 3D scene described in a computer?

3D Scene Representation

- Scene is usually approximated by 3D primitives
  - Point
  - Line segment
  - Polygon
  - Polyhedron
  - Curved surface
  - Solid object
  - etc.

3D Point

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

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

3D Vector

- Specifies a direction and a magnitude
- Represented by three coordinates
- Magnitude $|V| = \sqrt{dx^2 + dy^2 + dz^2}$
- Has no location

```c
typedef struct {
  Coordinate dx;
  Coordinate dy;
  Coordinate dz;
} Vector;
```
3D Vector

• Dot product of two 3D vectors
  \[ \mathbf{V}_1 \cdot \mathbf{V}_2 = ||\mathbf{V}_1|| \cdot ||\mathbf{V}_2|| \cdot \cos(\theta) \]

• Cross product of two 3D vectors
  \[ \mathbf{V}_1 \times \mathbf{V}_2 = \begin{vmatrix}
      \hat{x} & \hat{y} & \hat{z} \\
      dx_1 & dy_1 & dz_1 \\
      dx_2 & dy_2 & dz_2 
   \end{vmatrix} \]

3D Line Segment

• Linear path between two points

3D Ray

• Line segment with one endpoint at infinity
  \[ \mathbf{P} = \mathbf{P}_1 + t \mathbf{V}, \quad (0 \leq t \leq 1) \]

3D Line

• Line segment with both endpoints at infinity
  \[ \mathbf{P} = \mathbf{P}_1 + t \mathbf{V}, \quad (-\infty < t < \infty) \]
**3D Plane**
- A linear combination of three points

- Implicit representation:
  - \( P \cdot N + d = 0 \), or
  - \( ax + by + cz + d = 0 \)

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

**3D Polygon**
- Area “inside” a sequence of coplanar points
  - Triangle
  - Quadrilateral
  - Convex
  - Concave
  - Self-intersecting

Points are in counter-clockwise order
- Holes (use > 1 polygon struct)

**3D Sphere**
- All points at distance “\( r \)” from point “\((c_x, c_y, c_z)\)”
  - Implicit representation:
    - \( (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 = r^2 \)
  - Parametric representation:
    - \( x = r \cos(\phi) \cos(\Theta) + c_x \)
    - \( y = r \cos(\phi) \sin(\Theta) + c_y \)
    - \( z = r \sin(\phi) + c_z \)

**3D Scenes**
- Comprise set of geometric primitives

**Other Geometric Primitives**
- More detail on 3D modeling later in course
  - Point
  - Line segment
  - Polygon
  - Polyhedron
  - Curved surface
  - Solid object
  - etc.
3D Scene Example

This scene is a set of primitives

3D Scene Example

Simple scene composed of triangles

Overview

• 3D scene representation
  » 3D viewer representation
• Visible surface determination
• Lighting simulation

How is the viewing device described in a computer?

Camera Models

• The most common model is pin-hole camera
  ◦ All captured light rays arrive along paths toward focal point without lens distortion (everything is in focus)
  ◦ Sensor response proportional to radiance

Other models consider ...
  Depth of field
  Motion blur
  Lens distortion

View plane
Eye position (focal point)

Camera Parameters

• What are the parameters of a camera?

Camera Parameters

• Position
  ◦ Eye position (px, py, pz)
• Orientation
  ◦ View direction (dx, dy, dz)
  ◦ Up direction (ux, uy, uz)
• Aperature
  ◦ Field of view (xfov, yfov)
• Film plane
  ◦ "Look at" point
  ◦ View plane normal

Eye Position
"Look at" Point
View Plane
Up direction
View Plane

View Frustum

View Frustum Example

Overview

Visible Surface Determination

Ray Casting

• The color of each pixel on the view plane depends on the radiance emanating from visible surfaces

Simplest method is ray casting

• 3D scene representation
• 3D viewer representation
  » Visible surface determination
• Lighting simulation

How can the front-most surface be found with an algorithm?

• For each sample …
  ◦ Construct ray from eye position through view plane
  ◦ Find first surface intersected by ray through pixel
  ◦ Compute color of sample based on surface radiance
Ray Casting

• For each sample …
  - Construct ray from eye position through view plane
  - Find first surface intersected by ray through pixel
  - Compute color of sample based on surface radiance

Visible Surface Determination

• For each sample …
  - Construct ray from eye position through view plane
  - Find first surface intersected by ray through pixel
  - Compute color of sample based on surface radiance

Construct Ray

Ray: \( P = P_0 + tV \)

Ray Intersection Example

Ray Casting Example

Rays from camera in simple scene
Rendering

- Any samples can be used!
  - Rendering is a problem in sampling and reconstruction

Overview

- 3D scene representation
- 3D viewer representation
- Visible surface determination
  - Lighting simulation

Lighting Simulation

- Lighting parameters
  - Light source emission
  - Surface reflectance
  - Atmospheric attenuation
  - Camera response

Lighting Simulation

- Camera
- Surface
- Light Source

Lighting Example

- This scene is a set of primitives
Lighting Simulation

- Direct illumination
  - Ray casting
  - Polygon shading

- Global illumination
  - Ray tracing
  - Monte Carlo methods
  - Radiosity methods

More on these methods later!

Summary

- Major issues in 3D rendering
  - 3D scene representation
  - 3D viewer representation
  - Visible surface determination
  - Lighting simulation

- Concluding note
  - Accurate physical simulation is complex and intractable
    » Rendering algorithms apply many approximations to simplify representations and computations

Next Lectures

- Ray intersections
- Light and reflectance models
- Indirect illumination

For assignment #2, you will write a ray tracer!