CS 563 Advanced Topics in Computer Graphics
Camera Models

by Kevin Kardian
Introduction

- Pinhole camera is insufficient
  - Everything in perfect focus
  - Less realistic

- Different camera models are possible
  - Create varying images from the same scene
  - Simply generates rays differently
Camera Model Basics

- Each model stores very few values
  - Transforms between world and camera spaces
  - Distances to near and far clipping planes
  - Time values simulating camera shutter speed

- Coordinate Spaces
  - Object space
  - World space
  - Camera space
  - Screen space
  - Normalized device coordinate (NDC) space
  - Raster space
New Coordinate Spaces

• **Camera space**
  - Origin at the camera
  - z+ in the direction of viewing
  - y+ in the up direction

• **Screen space**
  - The camera space mapped to the image plane
  - z values are scaled to range [0,1]
  - These values correspond to points on the hither and yon planes
New Coordinate Spaces

• Normalized device coordinate (NDC) space
  • Scales all coordinates to range [0,1]
  • Note that y+ is in the down direction

• Raster space
  • Similar to NDC space
  • x,y coordinate values scaled to a different range
  • Based on the overall image resolution
Projective Models

- Subclass of normal camera models
- Projects objects from a space onto a screen
- Allows for depth of field
- Maintains several coordinate space transforms
  - CameraToScreen
  - WorldToScreen
  - RasterToCamera
  - ScreenToRaster
  - RasterToScreen
Orthographic Projection

- Projects a rectangular volume onto a screen
- Preserves parallel lines
- Maintains relative distance between objects
- Does not account for foreshortening
An Example

- Rendered using orthographic projection
- Note the lack of a vanishing point
- Platform edges remain parallel
Implementation Details

- Maps $z$ values to range of $[0,1]$
  - First, aligns $z = 0$ to the hither plane
  - Then, scales values so that $z = 1$ matches the yon plane
- Creating sample rays
  - Sample points are taken from raster space
  - The point is transformed to a point on the hither plane
  - The ray points straight down the $z$ axis
  - Finally, the ray is transformed to world space
Perspective Projection

- Projects a volume onto a screen
- This volume is not rectangular
- Does not maintain parallel lines
- Accounts for foreshortening
- More realistic view of object size and distance
- Rendered using perspective projection

- Note the illusion of a vanishing point

- Image appears to have depth
Implementation Details

• Plane of projection is actually at $z = 1$
  • One unit from the camera position

• Creating sample rays
  • Projecting sample points:
    • First, scale $z$ values to a range of $[0,1]$
    • Then, divide $x,y$ coordinate values by the scaled $z$
    • Finally, scale based on the fov angle to get $x,y$ coordinates to a range of $[-1,1]$

• Sample rays all point from the origin to this projection
• Actual lenses do not have perfect focus

• Circle of confusion
  • The image area onto which a single point is projected
  • Based on lens radius and focal distance
  • Focal distance - the distance at which the circle of confusion has no radius

• Large number of samples required for each pixel
Undersampling
Undersampling
• Get a random sample point on the lens

• Observation: Light through the center of a lens isn’t refracted
  • Generate this non-refracted ray
  • Find where it intersects the focal plane

• Sample ray originates at the sample point and points towards this intersection
Postprocessing Discussion

• Consider calculating the size of any give circle of confusion
• It is apparent that this can be done after ray tracing
• Each location on the scene can be “blurred” based on how focused it should appear
• Where are the flaws in this approach?
• How/why do these not apply to Cook et al’s approach?
Distributed Ray Tracing

Cook, Porter, Carpenter (1984)

- Approach was to achieve improvements by varying sample rays in time
- With extra samples, each spatial location could be sampled at several instants of time
- Instead, separate locations are sampled at varying times
- Oversampling still occurs
- Same result is achieved with fewer total samples
Motion Blur

Potmesil (1983)

- A preprocessing approach
- Attempting to render an image then apply blur is flawed
- Hidden surfaces may be revealed by motion
- What about background surfaces that are also in motion?
- What about other visual effects?
- Solution: Account for motion blur at the time of sampling
Motion Blur

Cook, Porter, Carpenter (1984)

- A distributed approach
- Different parts of an object are sampled at different times
- The object as a whole is captured in motion
- Accounts for various effects because their changes are captured as well
  - Visibility
  - Shading
  - Shadows
  - Depth-of-field
  - Reflections
• Images rendered using ray tracing have more flexibility

• Consider a point suspended in space

• Send rays in all directions from that point

• Scene maps to an image on a spherical plane

• Image manipulated to give a 2D view on a flat plane
An Example
• 180 deg. field of vision from top to bottom
• 360 deg. field of vision from left to right
• Note that this camera cannot use linear projections
  • There is no projection matrix

• Creating sample rays
  • All sample rays have the same origin
  • Sample points are converted to spherical coordinates
  • Coordinates are scaled to the appropriate ranges