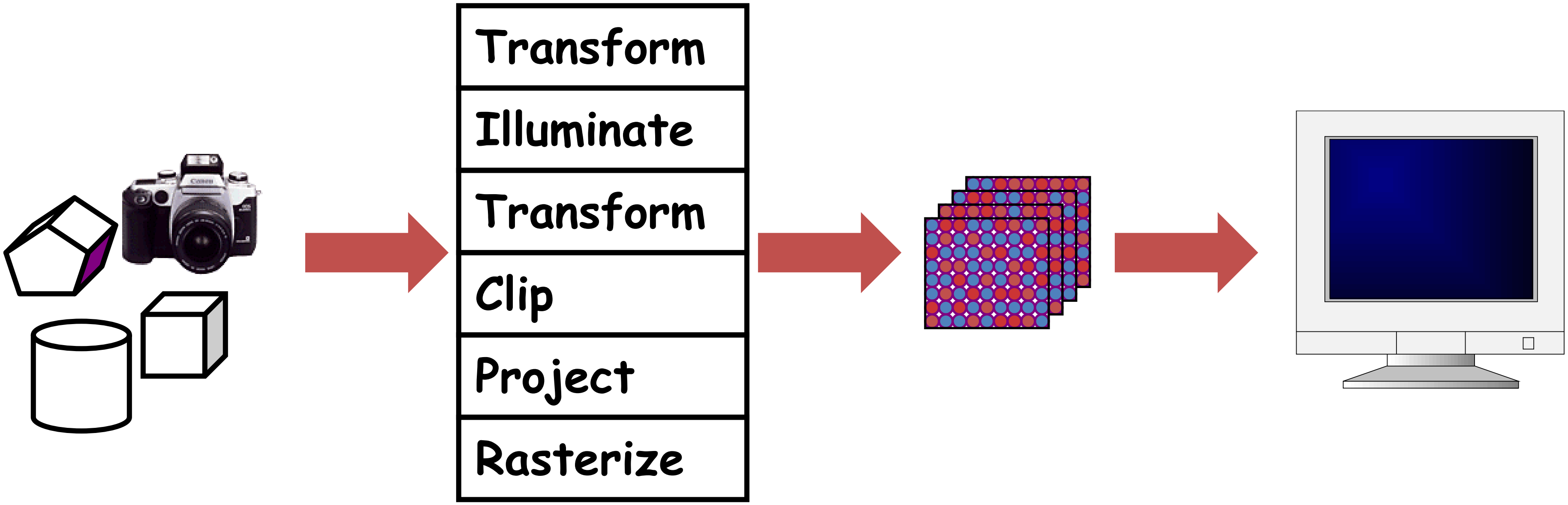
# 3-D Object representation
# &
# 3-D Transformation

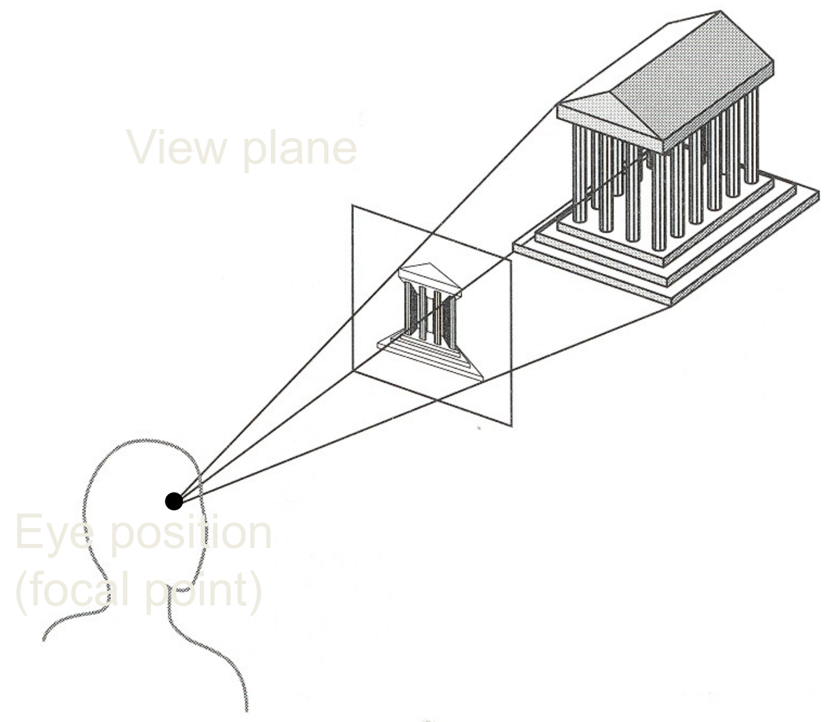
# Rendering 3D Scenes




Model & Camera Parameters | Rendering Pipeline | Framebuffer | Display

# Camera Models

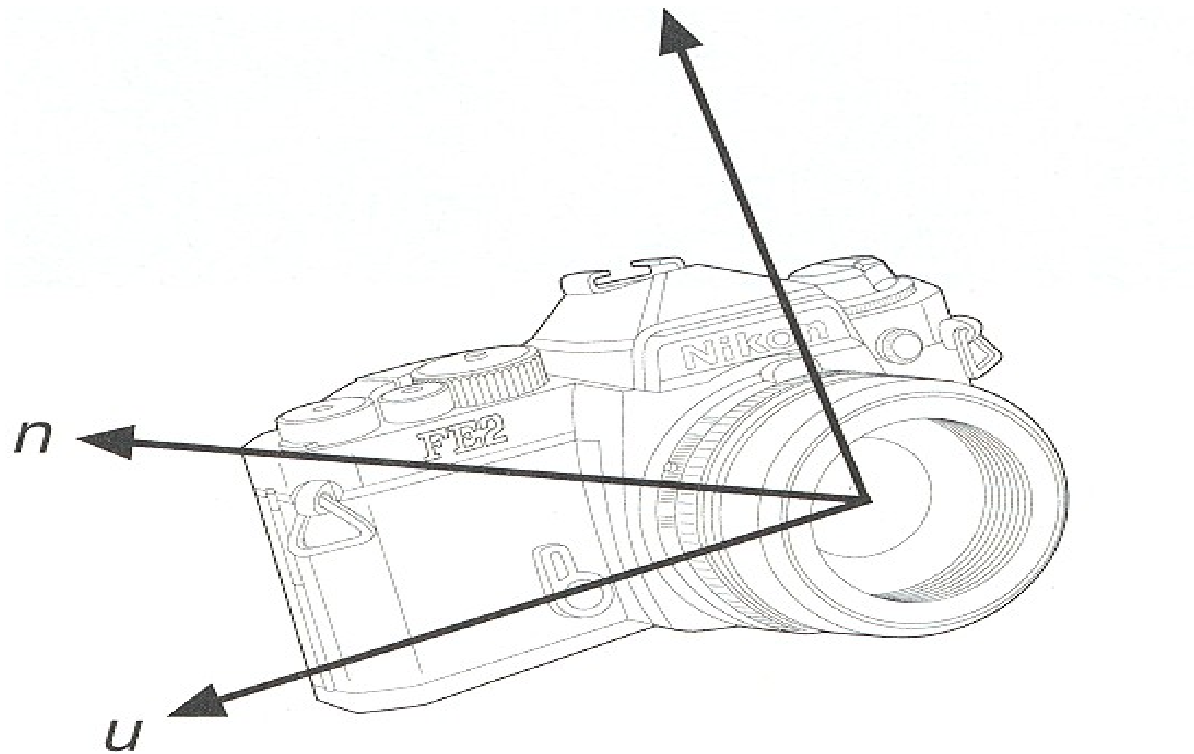
- The most common model is pin-hole camera
  - All captured light rays arrive along paths toward focal point without lens distortion (everything is in focus)
  - Sensor response proportional to radiance




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

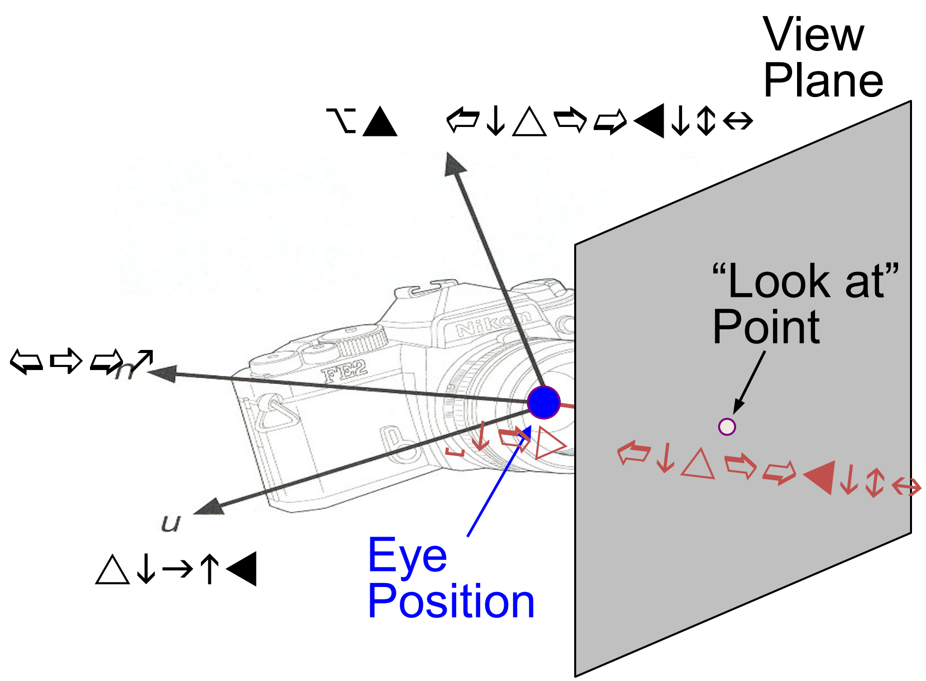
# Camera Parameters

- What are the parameters of a camera?

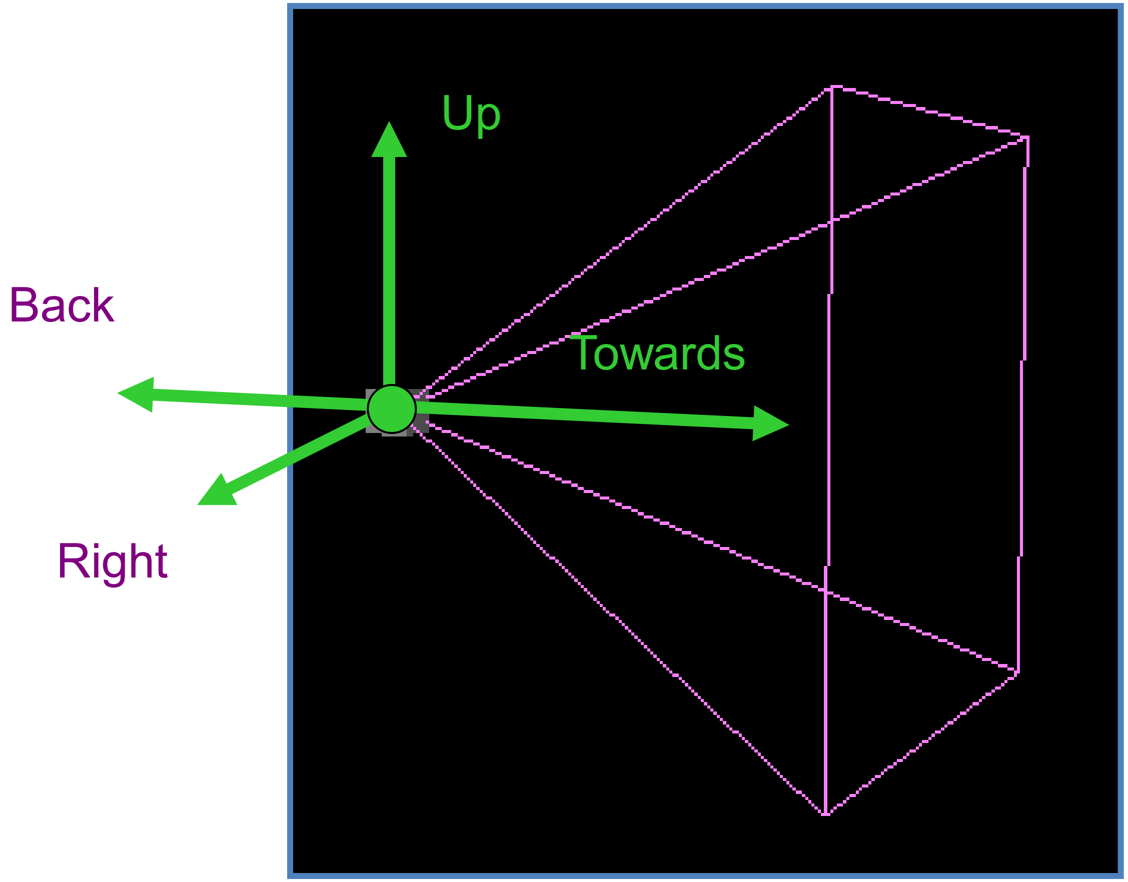


# Camera Parameters

- Position
  - Eye position (px, py, pz)
- Orientation
  - View direction (dx, dy, dz)
  - Up direction (ux, uy, uz)
- Aperture
  - Field of view (xfov, yfov)
- Film plane
  - “Look at” point
  - View plane normal

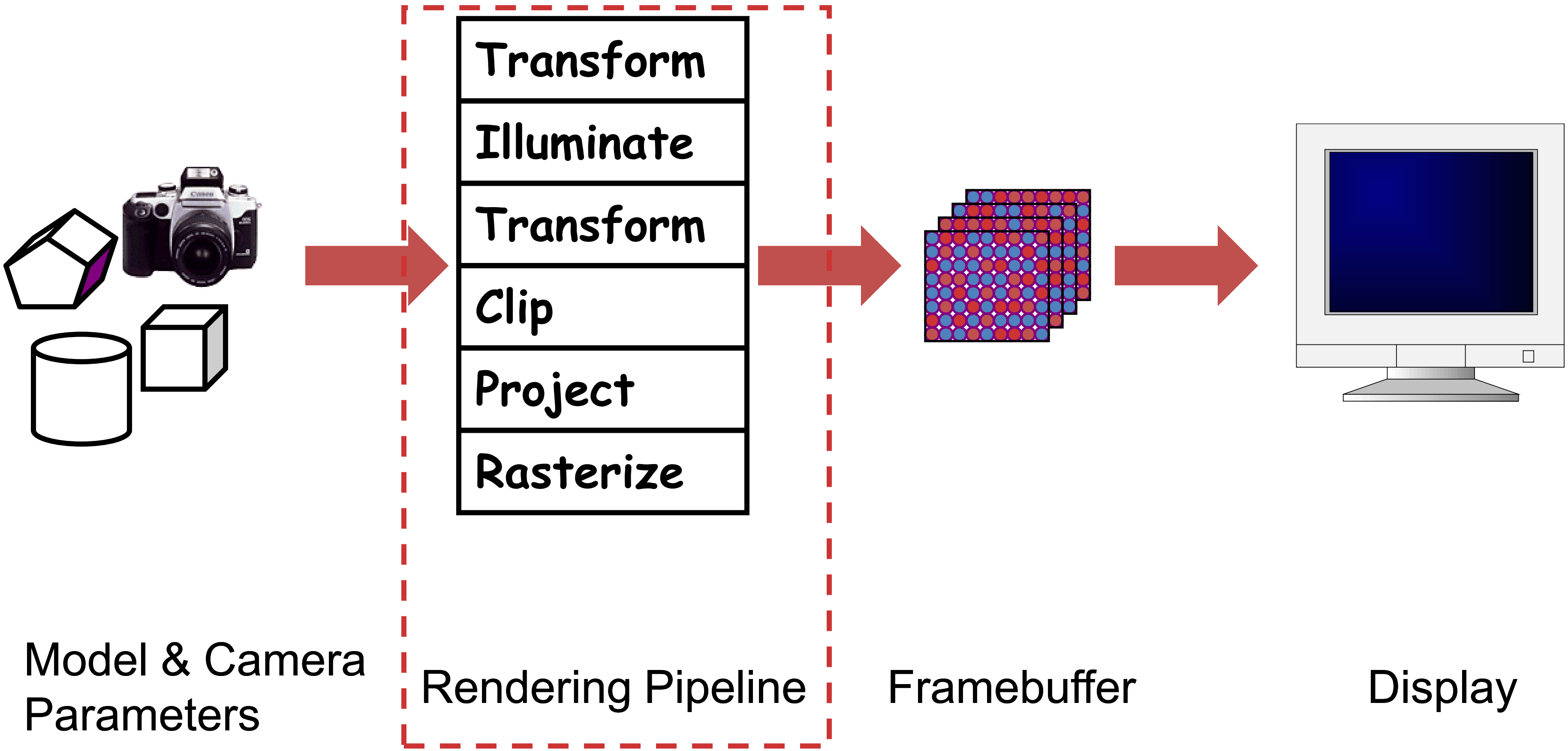


# Moving the camera




View Frustum

# The Rendering Pipeline



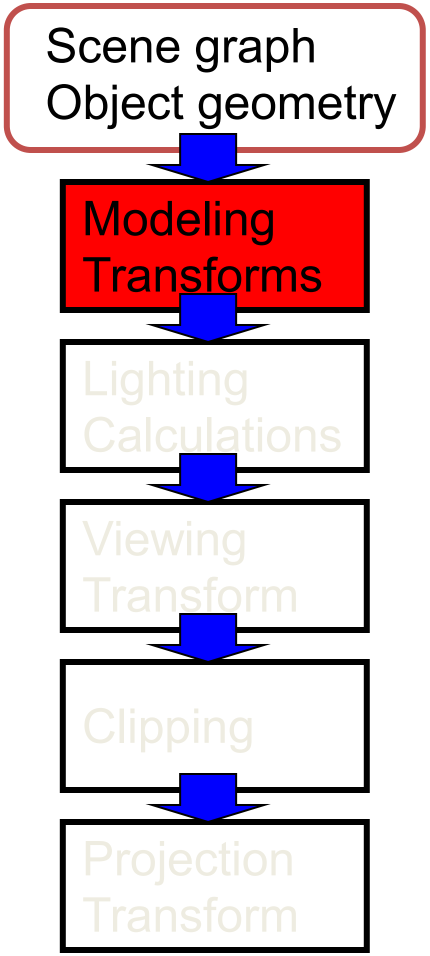
# Rendering: Transformations

- We’ve learned about transformations
- But they are used in three ways:
  - *Modeling transforms*
  - *Viewing transforms (Move the camera)*
  - *Projection transforms (Change the type of camera)*

# The Rendering Pipeline: 3-D

1. Scene graph
   Object geometry
2. *Modeling Transforms*
3. Lighting Calculations
4. *Viewing Transform*
5. Clipping
6. *Projection Transform*

# The Rendering Pipeline: 3-D

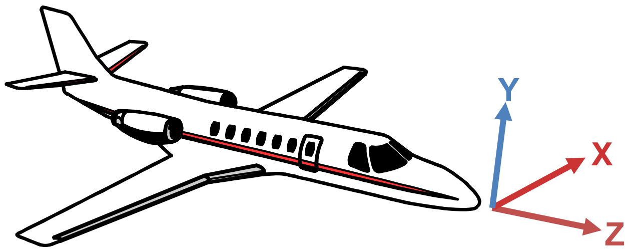



<u>Result:</u>

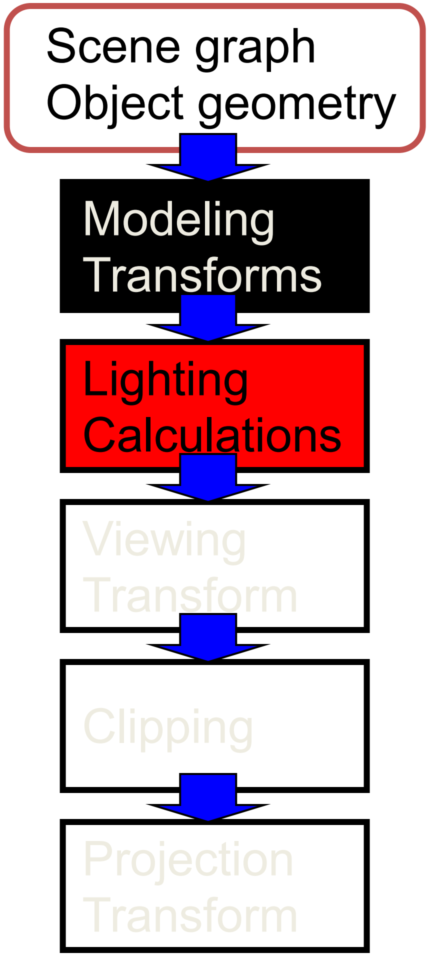
- All vertices of scene in shared 3-D “world” coordinate system

# Rendering: Transformations

- Modeling transforms
  - Size, place, scale, and rotate objects and parts of the model w.r.t. each other
  - Object coordinates -> world coordinates




# The Rendering Pipeline: 3-D

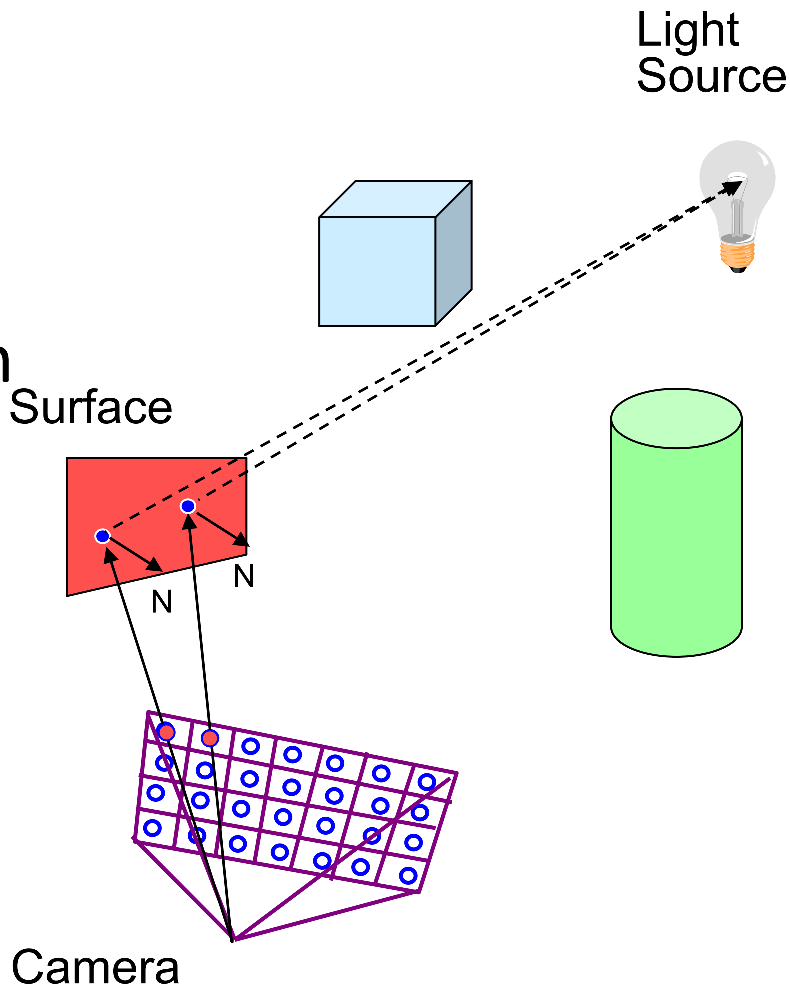



<u>Result:</u>

•All geometric primitives are illuminated

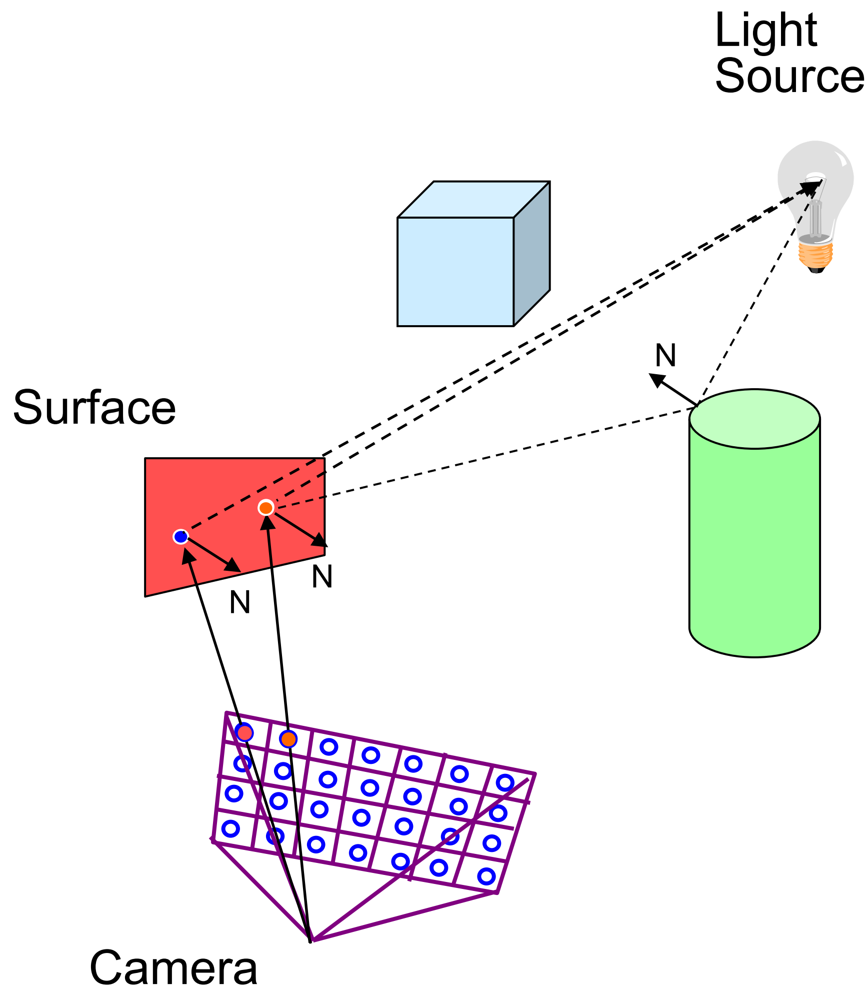
# Lighting Simulation

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



# Lighting Simulation

- Direct illumination
  - Ray casting
  - Polygon shading
- Global illumination
  - Ray tracing
  - Monte Carlo methods
  - Radiosity methods




More on these methods later!

# The Rendering Pipeline: 3-D

Scene graph
Object geometry

Modeling
Transforms

Lighting
Calculations

Viewing
Transform

Clipping

Projection
Transform

Result:

- Scene vertices in 3-D “view” or “camera” coordinate system

# Rendering: Transformations

- Viewing transform
  - Rotate & translate the world to lie directly in front of the camera
    - Typically place camera at origin
    - Typically looking down -Z axis
  - **World coordinates ⇨ view coordinates**

# The Rendering Pipeline: 3-D

Scene graph
Object geometry

Modeling
Transforms

Lighting
Calculations

Viewing
Transform

Clipping

Projection
Transform

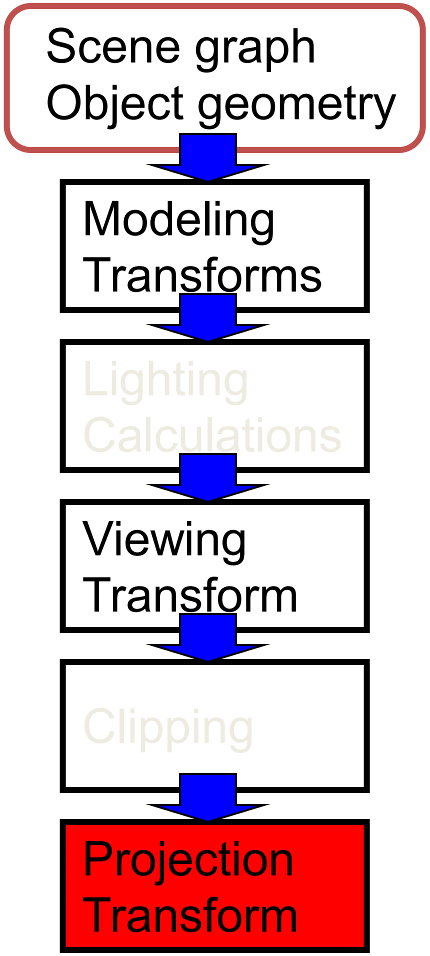
<u>Result:</u>

- Remove geometry that is out of view

# Assignment 2

- Due two and a half weeks from today
  - Project description available online
  - We’ll discuss details in class on Monday

# The Rendering Pipeline: 3-D

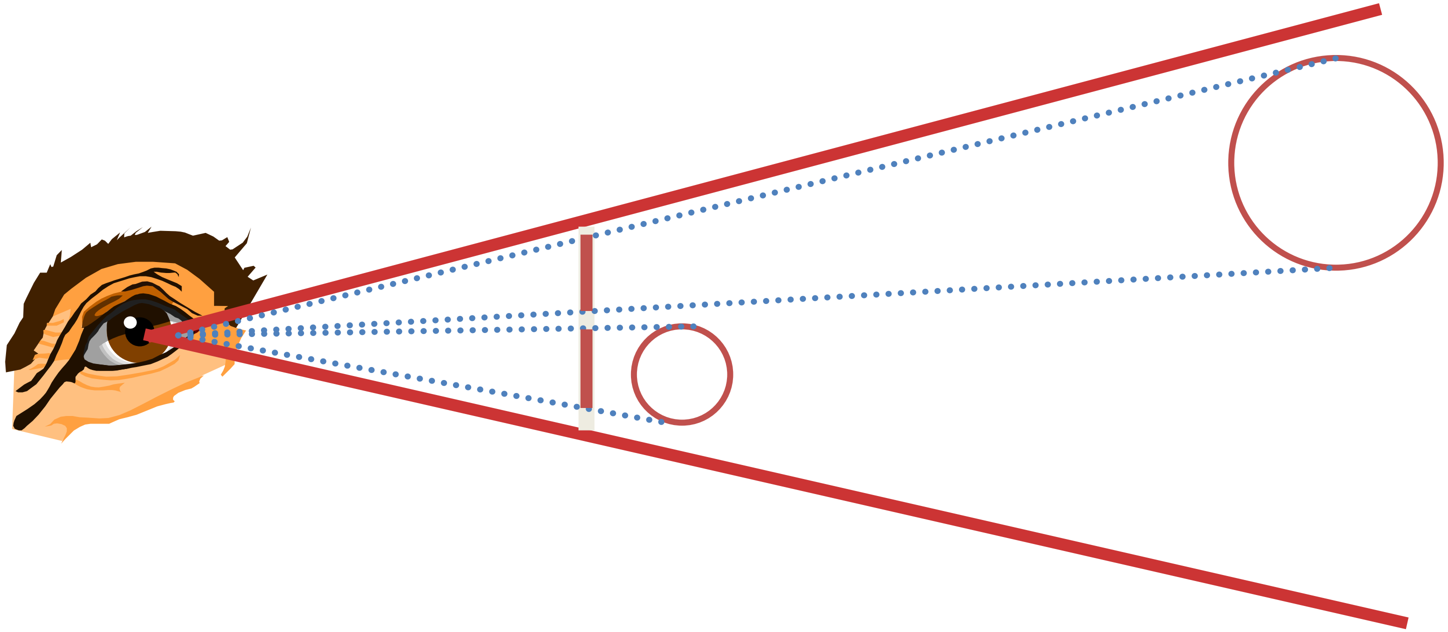



<u>Result:</u>

- 2-D screen coordinates of clipped vertices

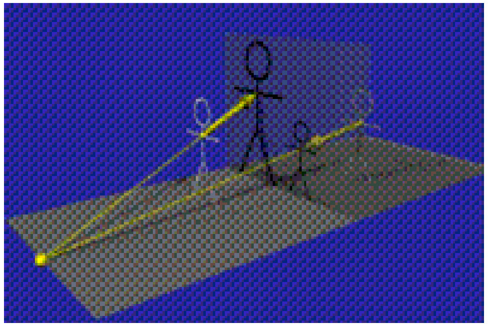
# Rendering: Transformations

- Projection transform
  - Apply perspective foreshortening
    - Distant = small: the pinhole camera model
  - View coordinates ⇨ screen coordinates

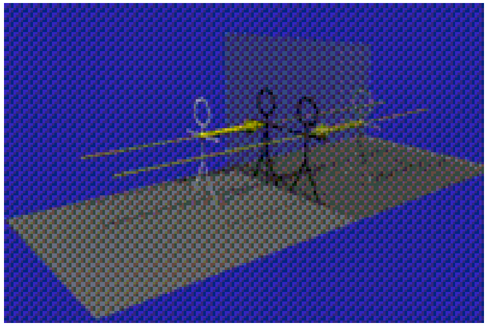


# Rendering: Transformations

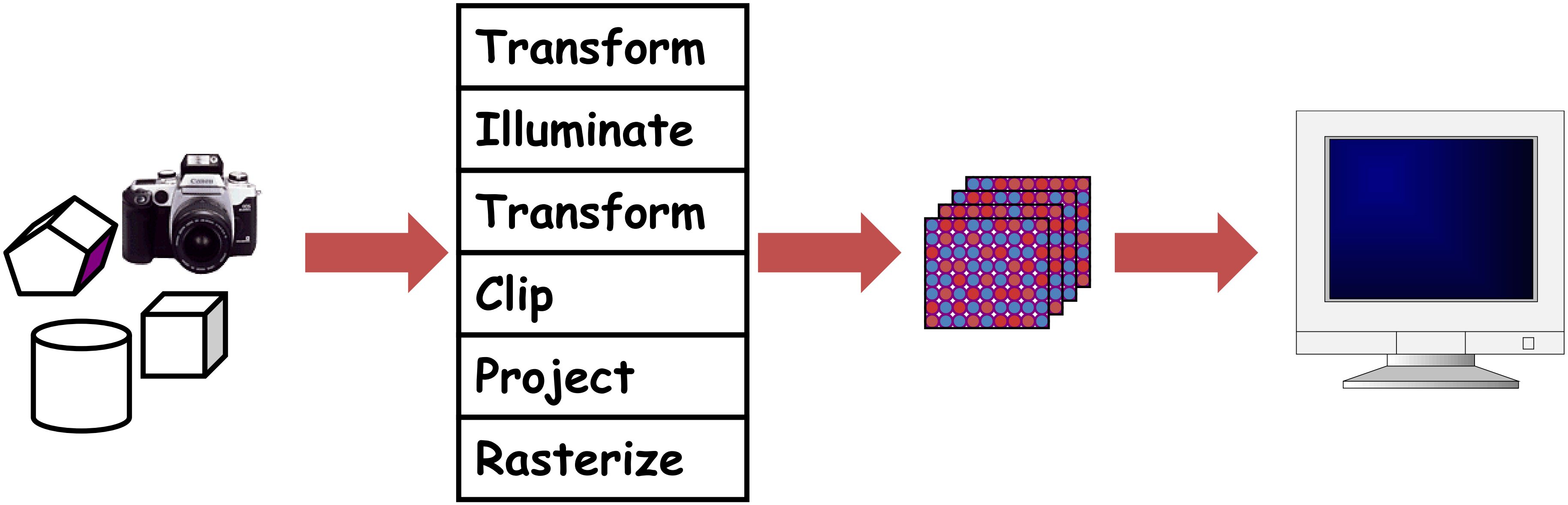
- Perspective Camera



- Orthographic Camera



# Rendering 3D Scenes




Model & Camera Parameters

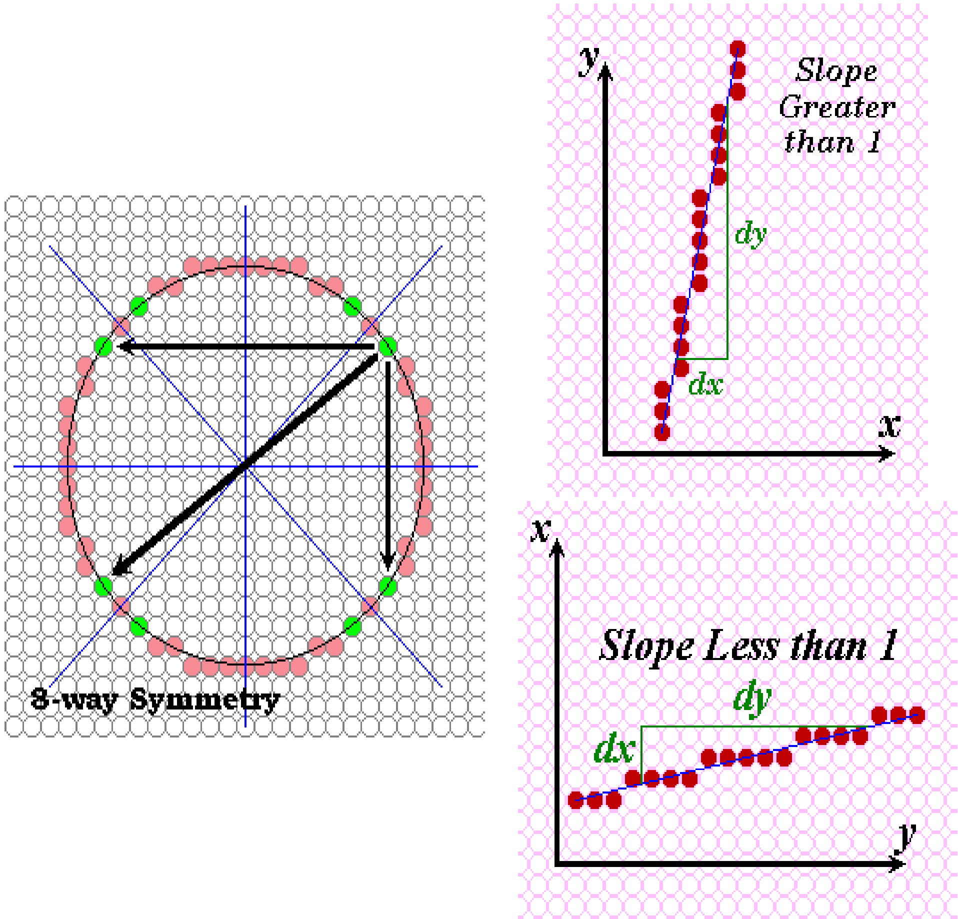
Rendering Pipeline

Framebuffer

Display

# Rasterize

- Convert screen coordinates to pixel colors



# Summary

- Geometric primitives
  - Points, vectors
- Operators on these primitives
  - Dot product, cross product, norm
- The rendering pipeline
  - Move models, illuminate, move camera, clip, project to display, rasterize