CS155b – Computer Graphics

Instructor: Giovanni Motta (gim@ieee.org)
Volen, Room #255. Phone: x62718

Class:

Mon. and Wed. from 5 to 6:30pm
Abelson #131

Teaching Assistants:

Anthony Bucci (abucci@cs)
John Langton (psyc@cs)
Anurag Maskey (anurag@cs)

Books

Textbook:

Computer Graphics: Principles and Practice in C, by J. D. Foley, A. Van Dam, S. K. Feiner, J. F. Hughes. Addison-Wesley, 2nd ed..

OpenGL:

OpenGL Programming Guide: The Official Guide to Learning OpenGL, Version 1.2, by M. Woo, J. Neider, T. Davis, D. Shreiner, OpenGL Architecture Review Board. Addison-Wesley, 3rd ed..

Suggested:

Mathematics for 3D Game Programming & Computer Graphics, by Eric Lengyel. Charles River Media.

Additional References

Web Page:

http://www.cs.brandeis.edu/~cs155

Lectures:

Published on the web page in Adobe pdf format.

Demo, Sample Programs, Useful Links:

Web page.

Essential Math Reference Book:

Essential Mathematics for* Computer *Graphics, fast,
by John Vince. Springer.

Homework

Programming:
With OpenGL library called from C/C++.

Theory:
Will cover the topics discussed in class.

In general, two weeks due date.
Solution will be given in class on due date.
No late homework accepted.

Exams:
Midterm and Final. In class, closed book.

Goals

Learning the principles of Computer Graphics

Understanding graphical models, fundamental techniques, algorithms and implementation issues

Practicing some applied mathematics

Getting acquainted with a Graphical Library (OpenGL)

Syllabus

Introduction

Overview, Applications, Examples.

2D Drawing

Scan Conversion of Lines and Circles, Polygon Clipping, Polygon Filling.

2D Viewing and Geometrical Transformations

Rotation, Reflection, Shear, Scale and Translation. World to Viewport Coordinate Transformation.

3D Solid Modeling

3D Models and Representations, Curves and Surfaces.

3D Viewing and Geometrical Transformations

Geometrical Transformations, Projections and Viewing in 3D, Visible Surface Algorithms.

Color

Color Spaces, Metrics, Transformations.

Illumination and Shading

Light Models, Shading Models, Transparency, Shadows.

Free Form Modeling

Interpolation and Approximation, Curve and Surface Splines.

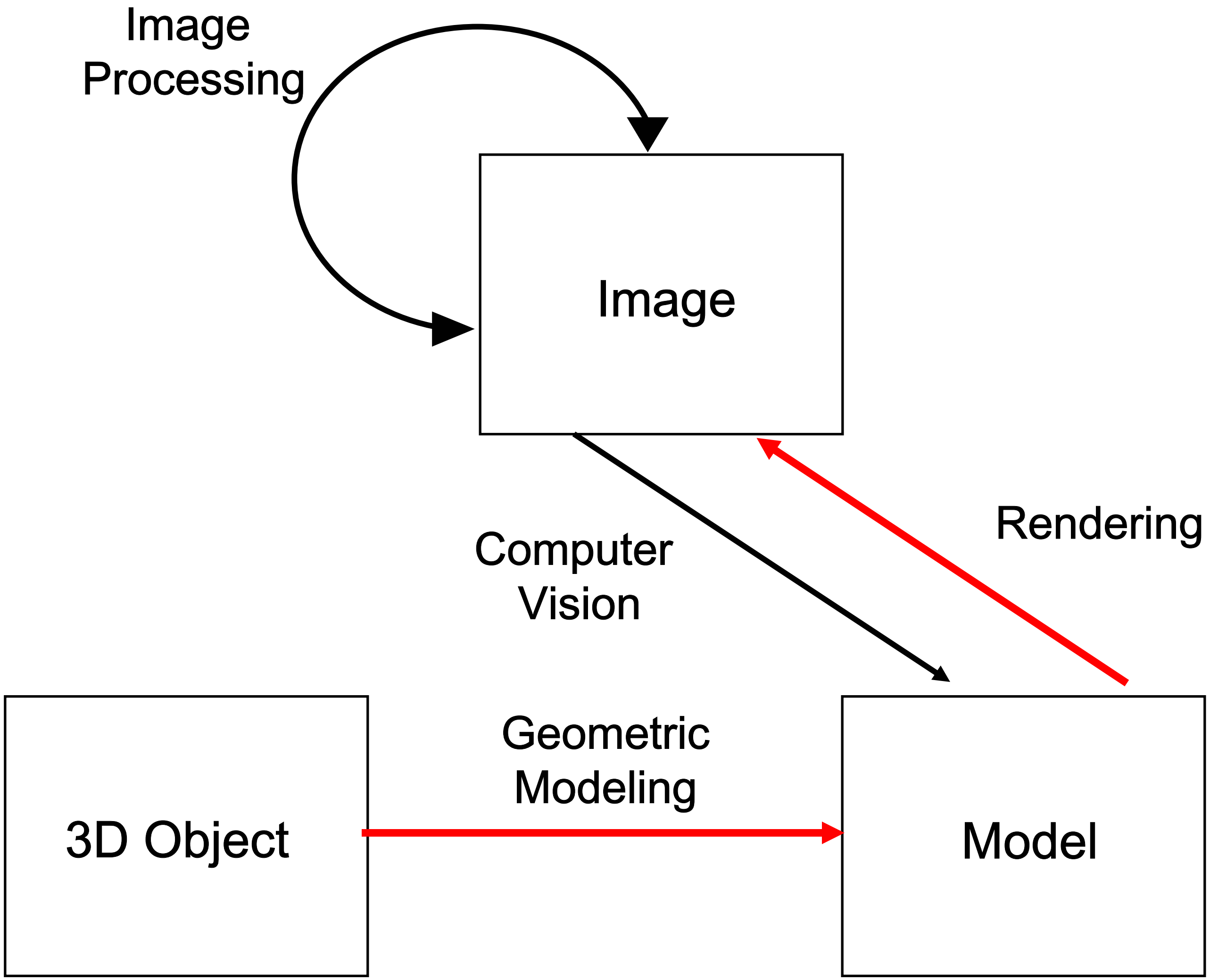
Advanced Topics

Ray Tracing, Texture Mapping, Animation, Morphing, Physics Based Models.

Applications

- CAD - Computer Aided Design (Mechanical, Architectural)
- Simulators (Flight, Driving, Sports)
- Advertising
- Virtual Reality
- Architectural Visualization
- Art and Entertainment
- Games
- Special effects
- Education
- Scientific visualization

The Visual Sciences



The Visual Sciences

Image Processing:

From Images to Images

Computer Vision:

From Images to Models

Computer Graphics:

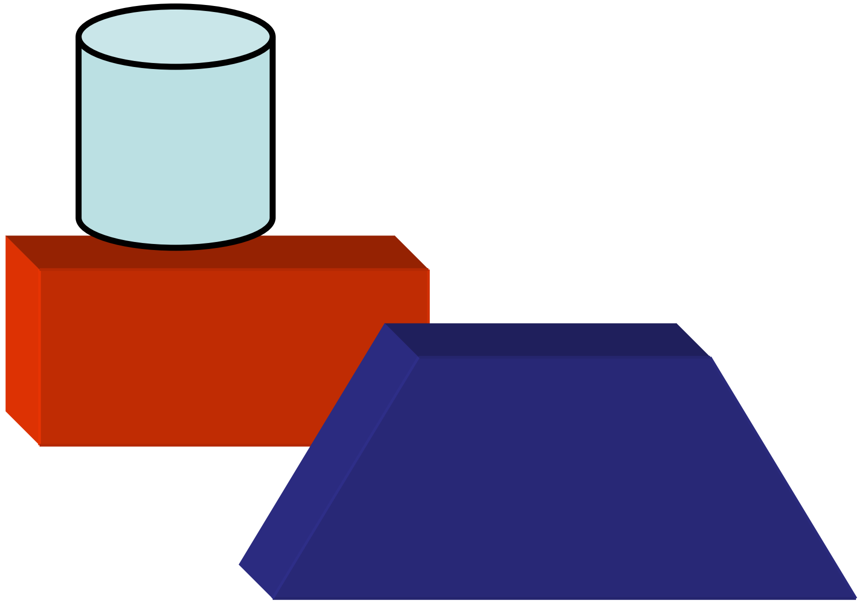
From Objects to Models (Geometric Modeling).

From 2D/3D Models to Images (Rendering).

From 4D Models to Images (Animation).

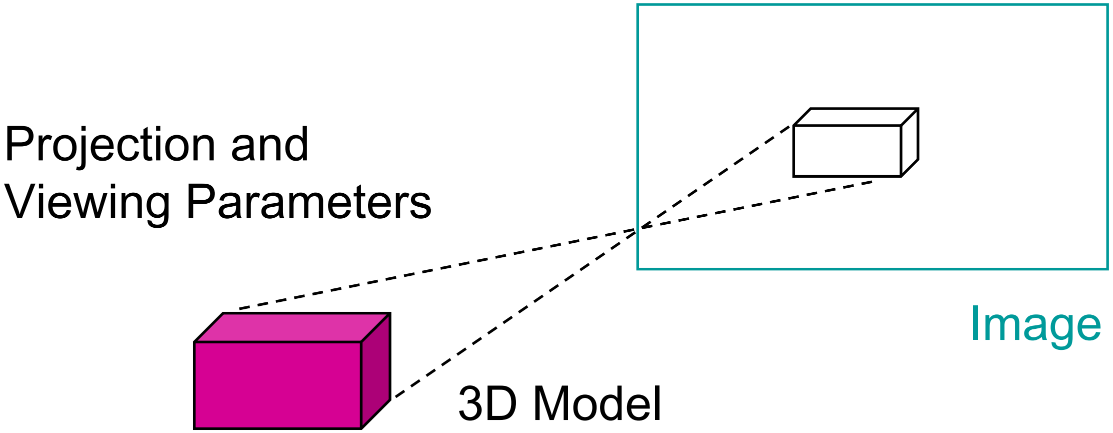
Geometric Modeling

- From a concept (or a real object) to a geometric representation on a computer
- Example: a sphere can be described as (*x*,*y*,*z*,*r*)
- Complex objects can be constructed from simpler ones



Rendering

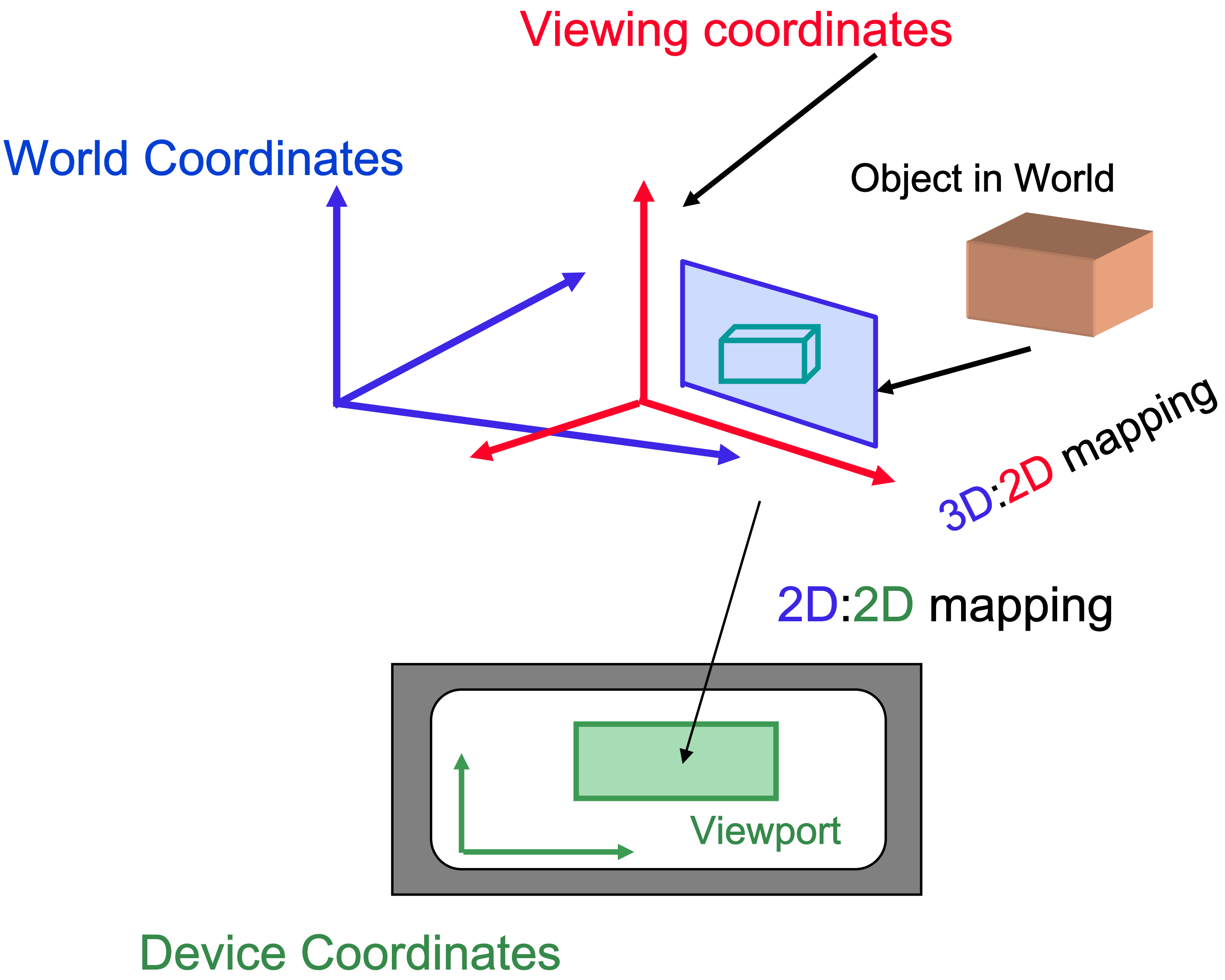
- Given a scene and viewing parameters, produce an image
- Images are a 2D array of pixels
- Important sub problems:
  - –Which pixels are covered by each object ? (Scan Conversion)
  - –What is visible at each pixel ? (Visible Surface Algorithm)
  - –What color should a pixel be ? (Illumination, Shading Algorithms).



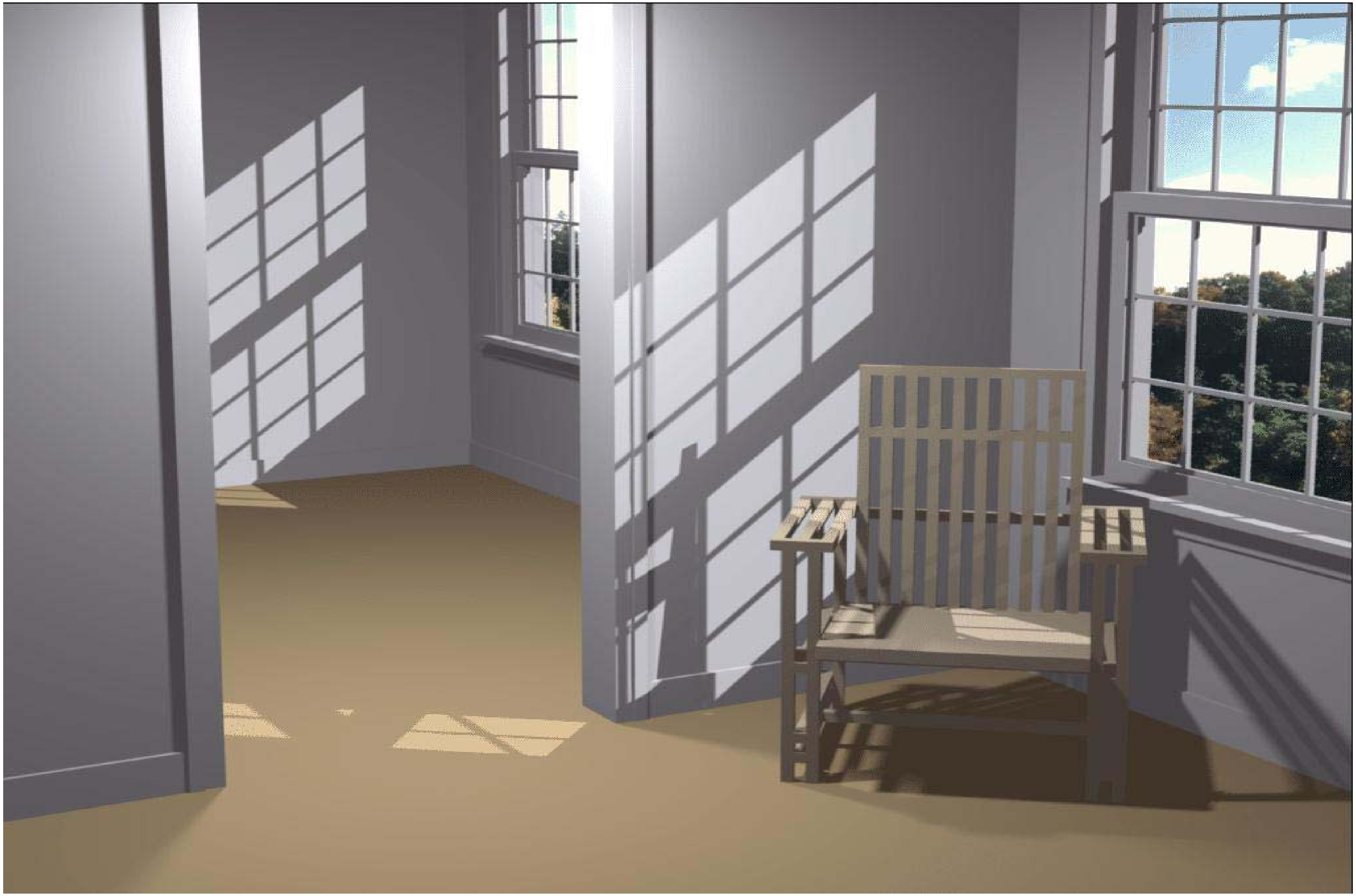
Animation

- Definition of complex time-dependent behavior of objects
- Issues with rigid and elastic joints
- Realistic rendering of collective behaviors
- Examples:
  - Automatic interpolation between key-frames
  - Physics based simulation

Viewing Transformation Pipeline



Rendered Image



Viewing Factors

- **Objects:**
  - Geometrical Properties of an Object (Solid Modeling)
  - Physical Properties of Object's Surfaces (Illumination Models, Color Models)
- **Camera:**
  - Projections
- **Light Source:**
  - Color Theory
- **Spatial set-up:**
  - 3D Transformations, Coordinate Systems

2D Drawing

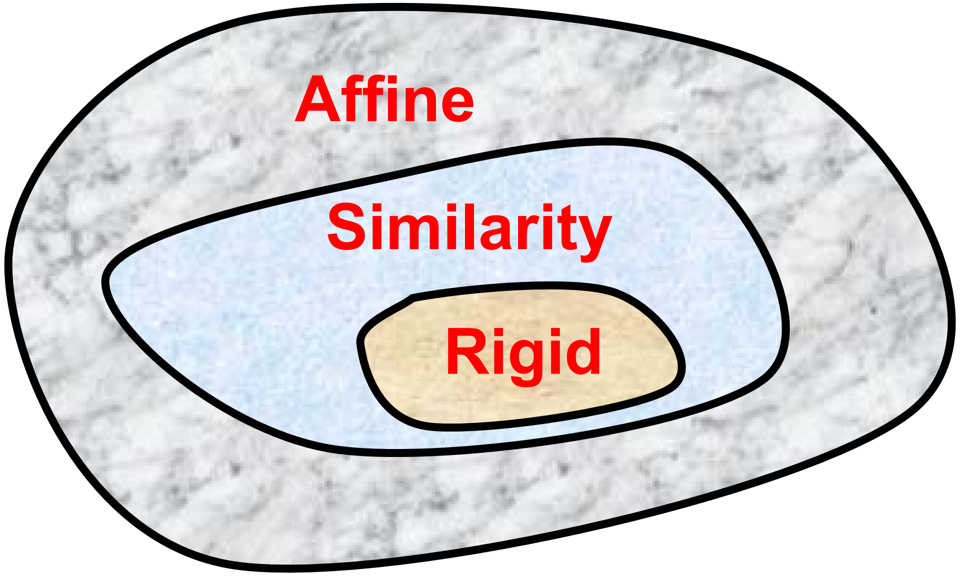
Goal: Getting Acquainted with Images

- Displays (Raster vs. Vector)
- Basic Definitions: Pixel, Resolution, Dynamic Range…
- Line Drawing (Incremental and Mid-Point Algorithms)
- Techniques for Drawing Circles
- Filling Polygons

2D Transformations

Goal: Introduction to 3D, Review Linear Algebra

- Basic 2D Transformations: Translation, Scaling, Rotation, Shear.
- Composition of Transformations and Transformation Groups:



2D Transformations

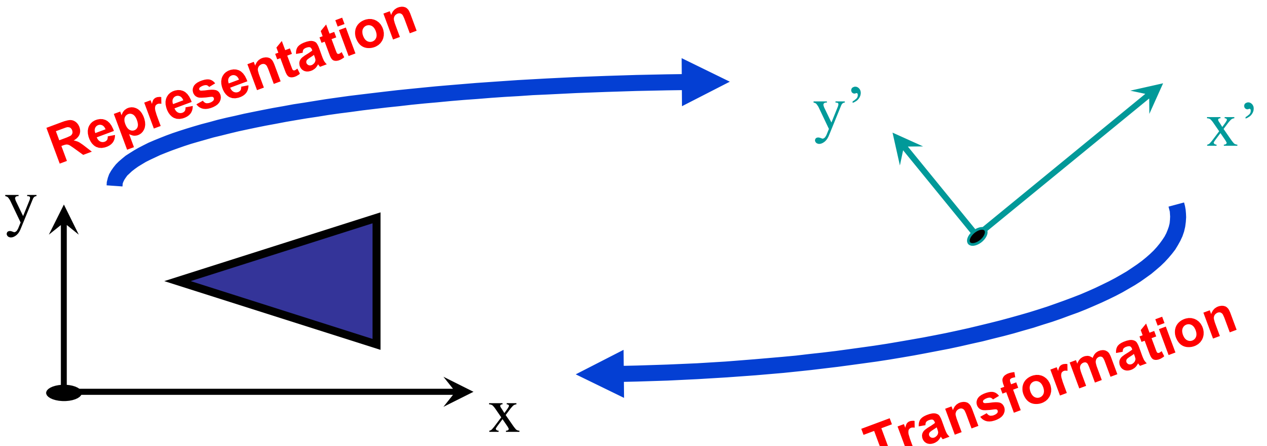
- Transformations in Matrix notation:

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

- Composition of transformations in matrix notation
- The homogeneous coordinates in 2D:

$$(x, y) \rightarrow (X, Y, W) = (tx, ty, t)$$

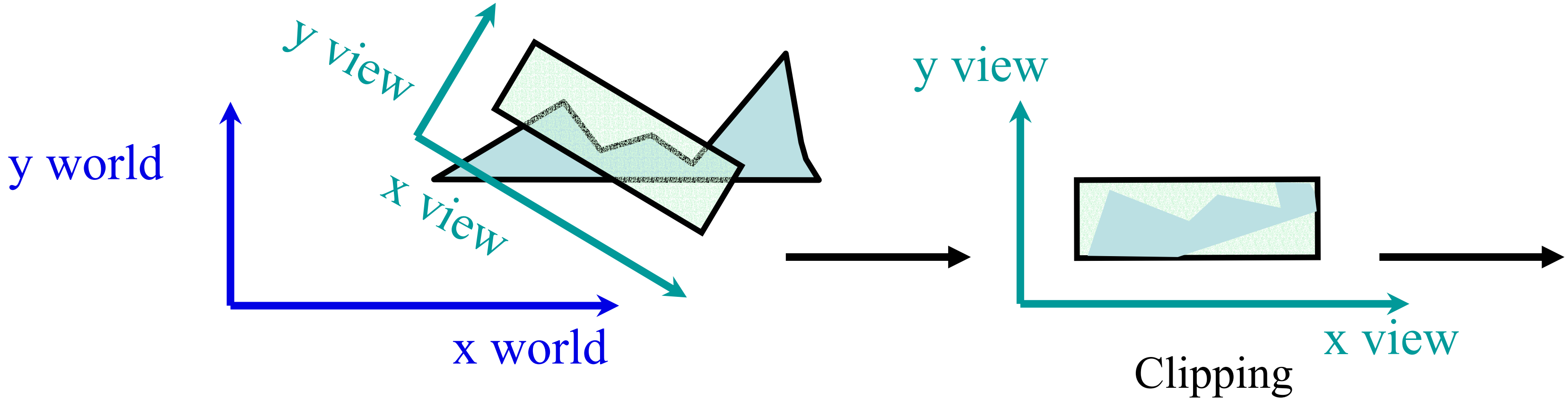
- Change of coordinates:



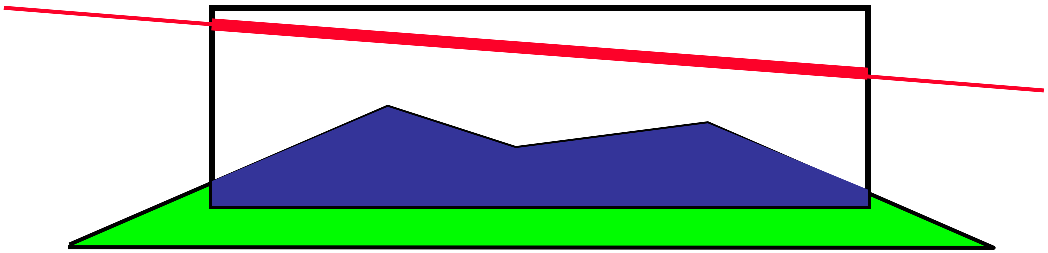
2D Viewing

Goal: Introduction to 3D and some Rendering Concepts

Viewing Transformation pipe-line:



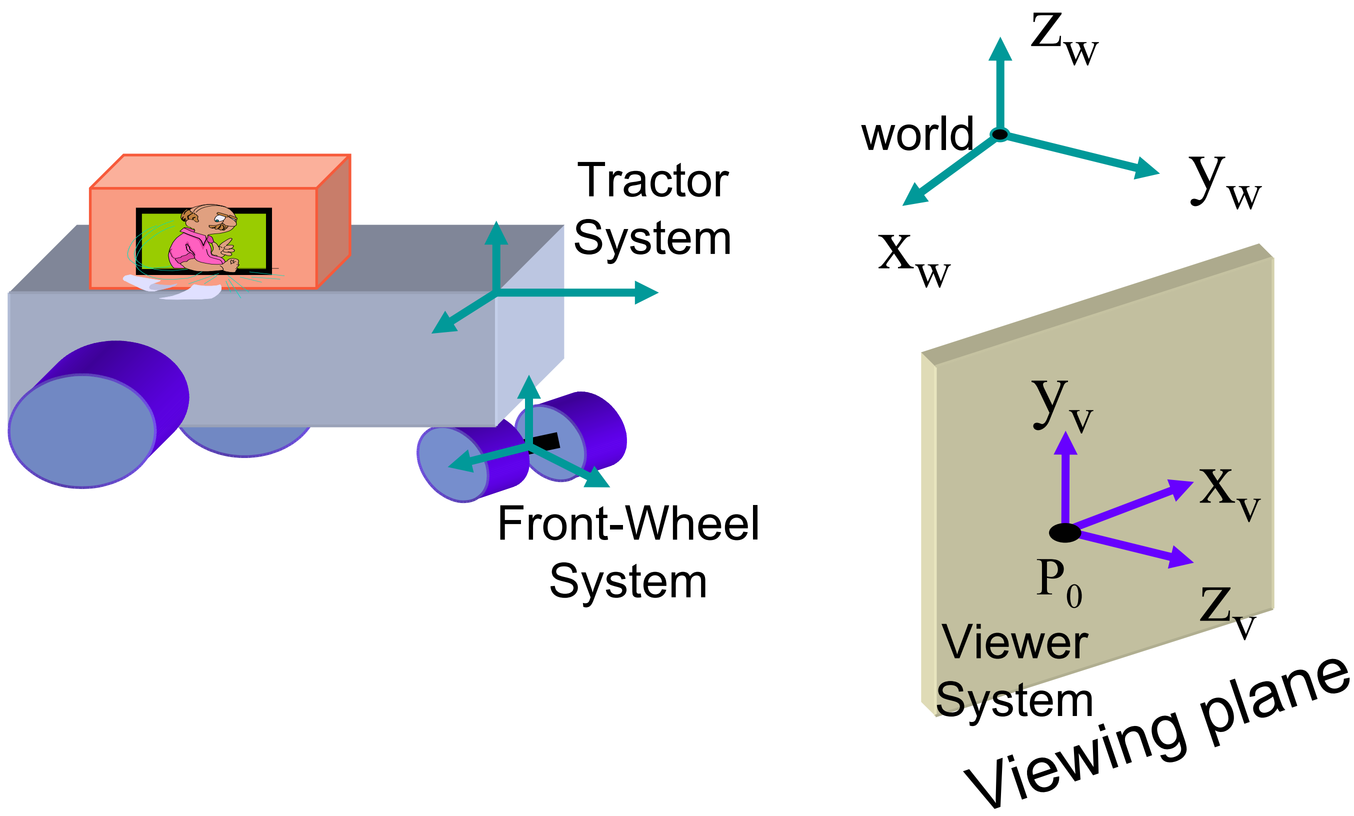
Line and Polygon Clipping:



3D Viewing

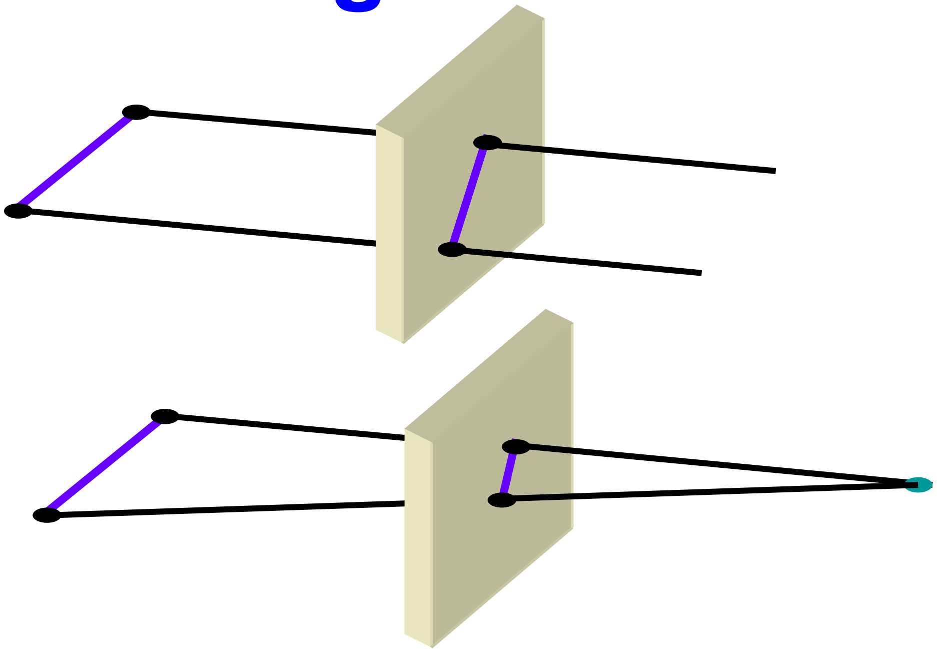
Goal: Geometrical Transformations in Viewing Pipe-iine

From Model Coordinates to Viewer Coordinates:

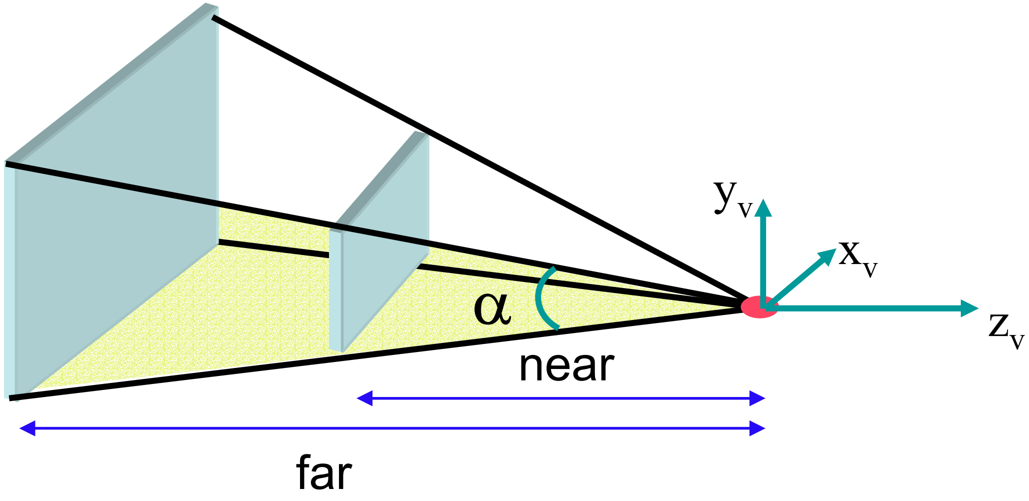


3D Viewing

- Projections:
  - –Orthographic
  - –Oblique
  - –Perspective



- The Viewing Volume:

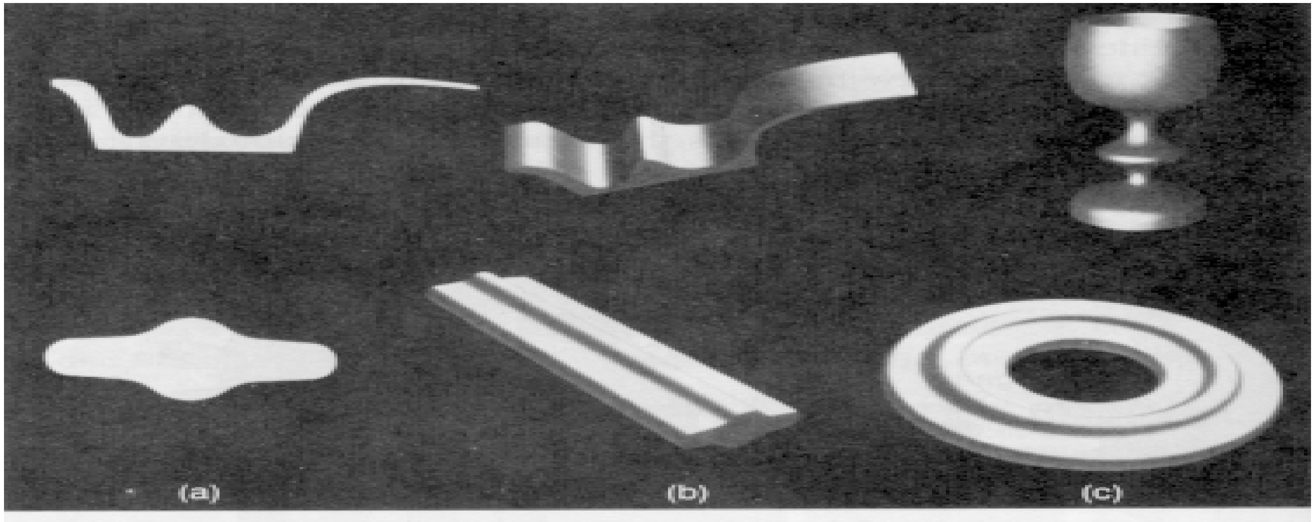


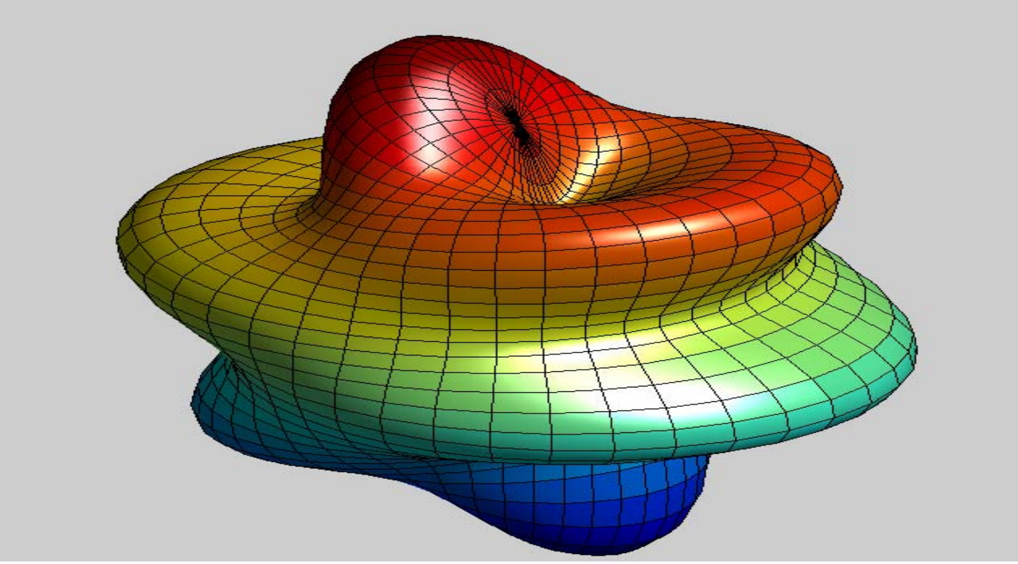
Solid Modeling

Goal: Learn how to Define Solid Objects

- **1D Curves in 3D**
  - Primitive based: line segments.
  - Free form:
    - Implicit, Explicit, Parametric (Polynomials, Splines)
- **2D Surfaces in 3D**
  - Primitive Based: Polygon Mesh
  - Free Form: As Above
- **3D volumes in 3D**
  - Volume Rep.
    - Sweep Volumes
    - Spatial Occupancy (Voxels, Octree, …)
    - Constructive Solid Geometry
  - Boundary Rep.
    - Polyhedra
    - Free Form: As Above

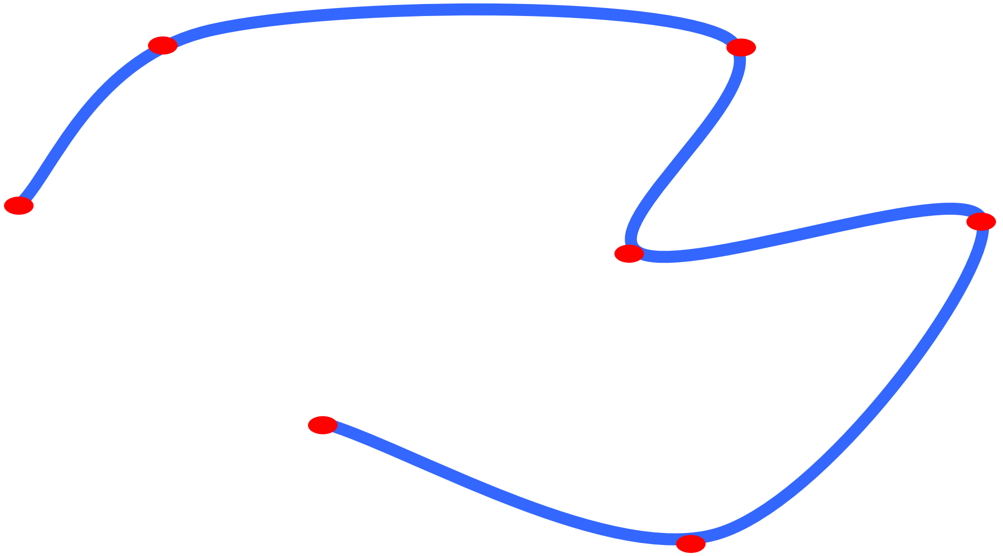
Solid Modeling





Parametric Surfaces

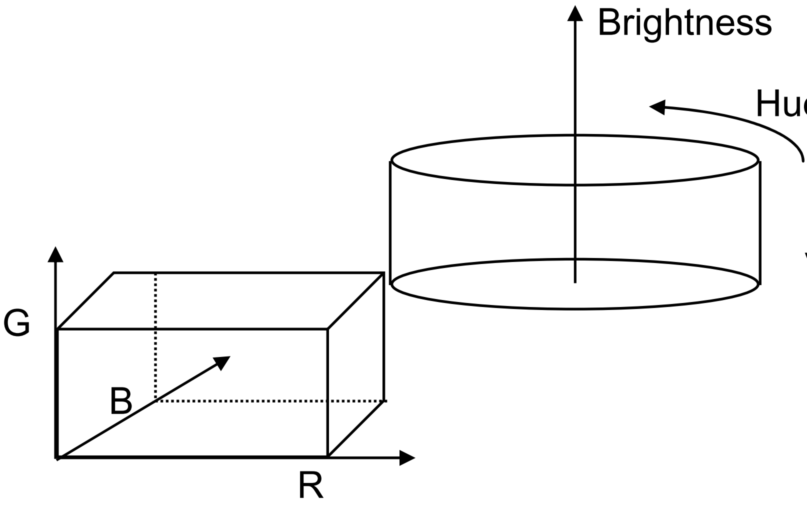
- **Bilinear Interpolation**
- **Splines:**
  - Cardinal Spline
  - Hermite Spline
  - Bezier Spline
  - B Spline

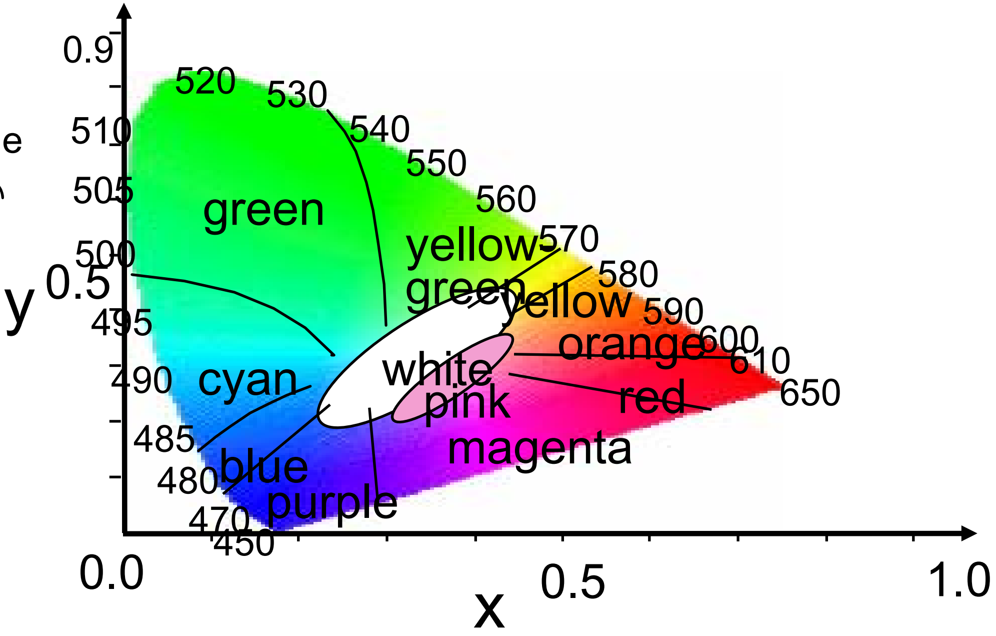


Color Theory

Goal: Understanding what a color is

- The Trichromatic Color Theory
- Linear Color Space and Color Representations: RGB, CMY,HSB
- Perceptual Color Spaces: LAB,YIQ
- The CIE Chromaticity Diagram

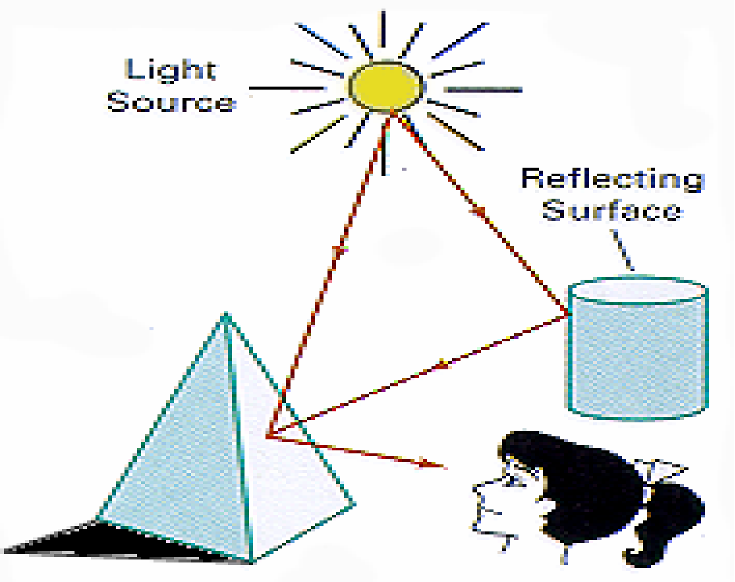




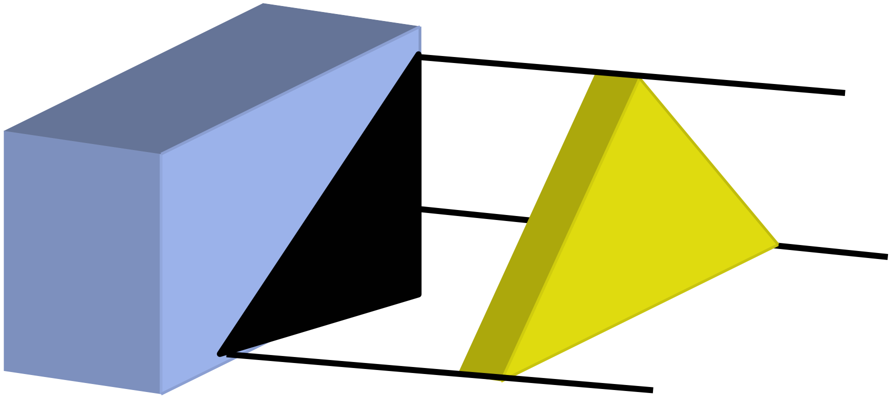
Illumination Models and Shading

Goal: Understanding the physical properties of an object

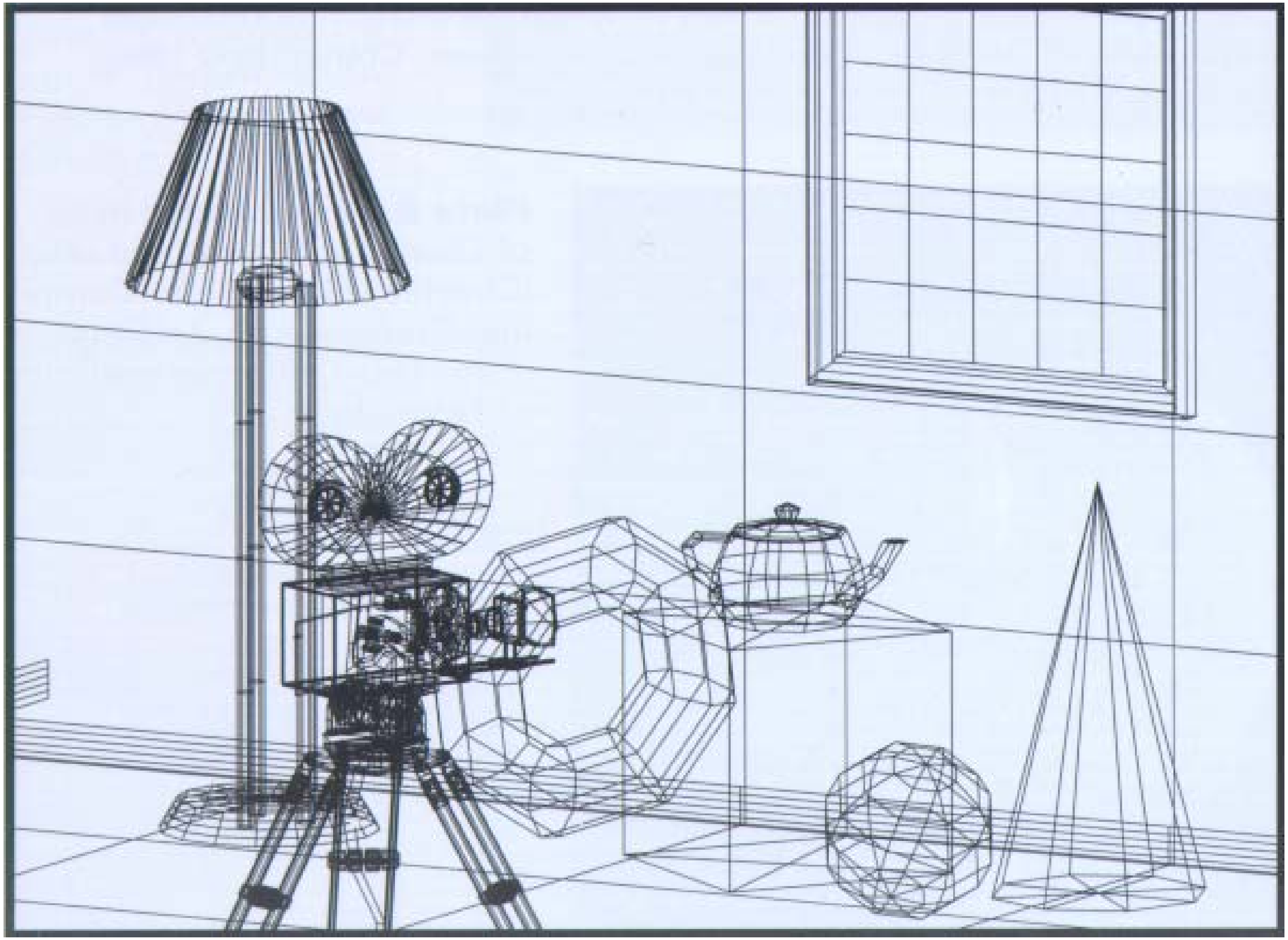
- Light Source Parameters (Shape, Position, Color, Intensity)
- Surface Parameters: Ambient, Diffuse, Specular
- Polygon Rendering Methods
- Transparency
- Shadow



Illumination Models and Shading

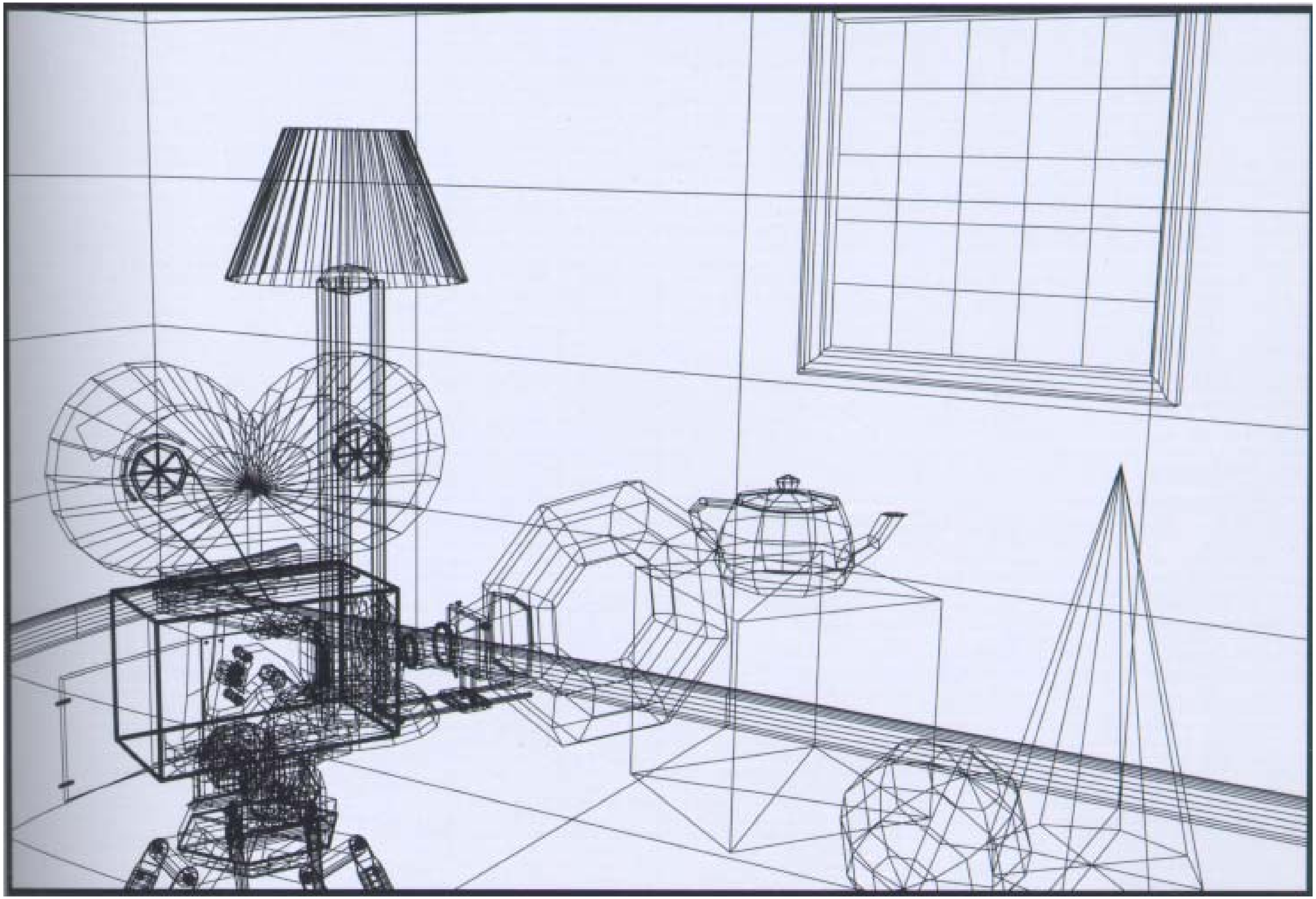


Example: Creating an Image from a Model



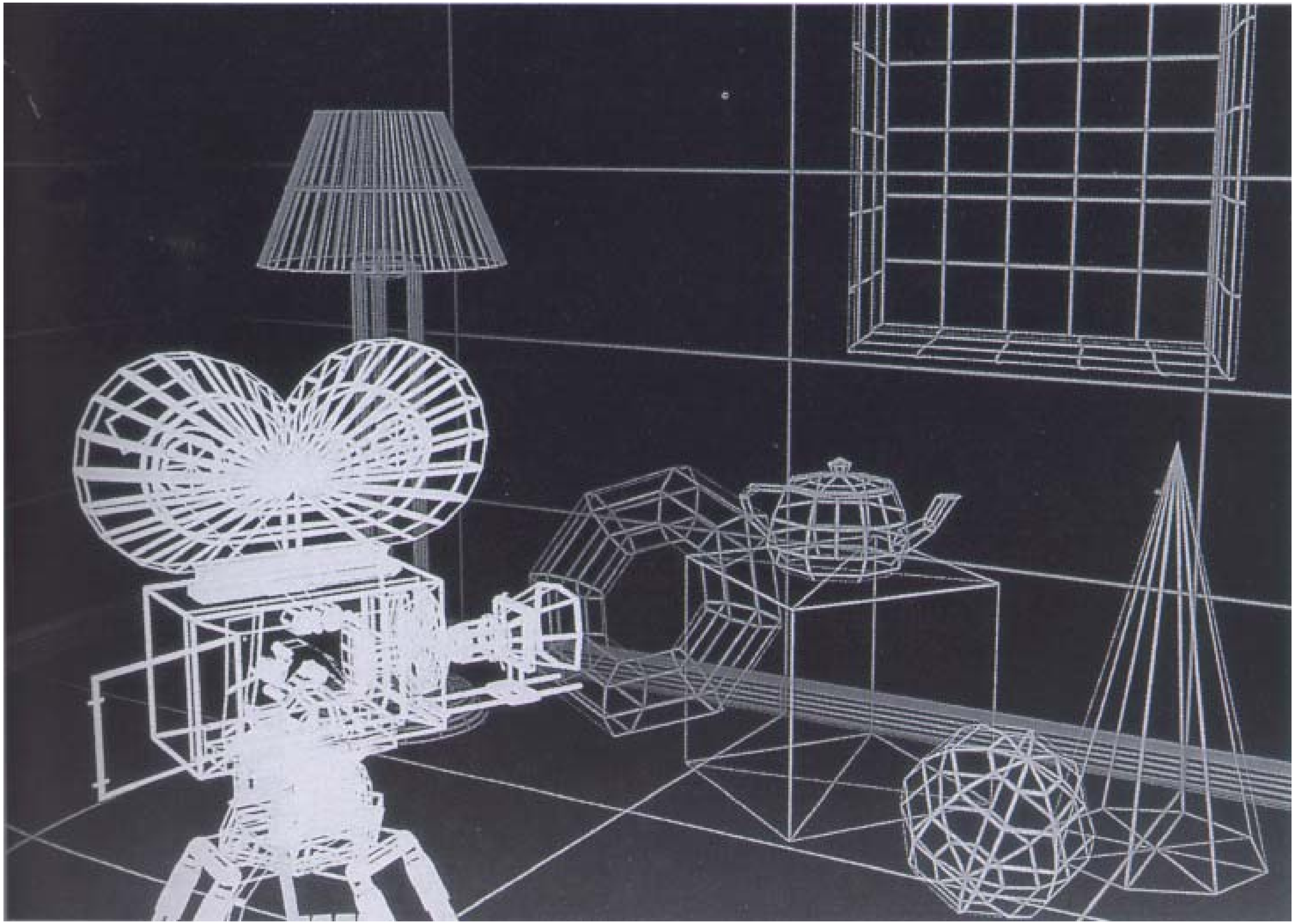
Polygonal Model Generated from Spline Patches. Orthographic Projection

Example: Creating an Image from a Model



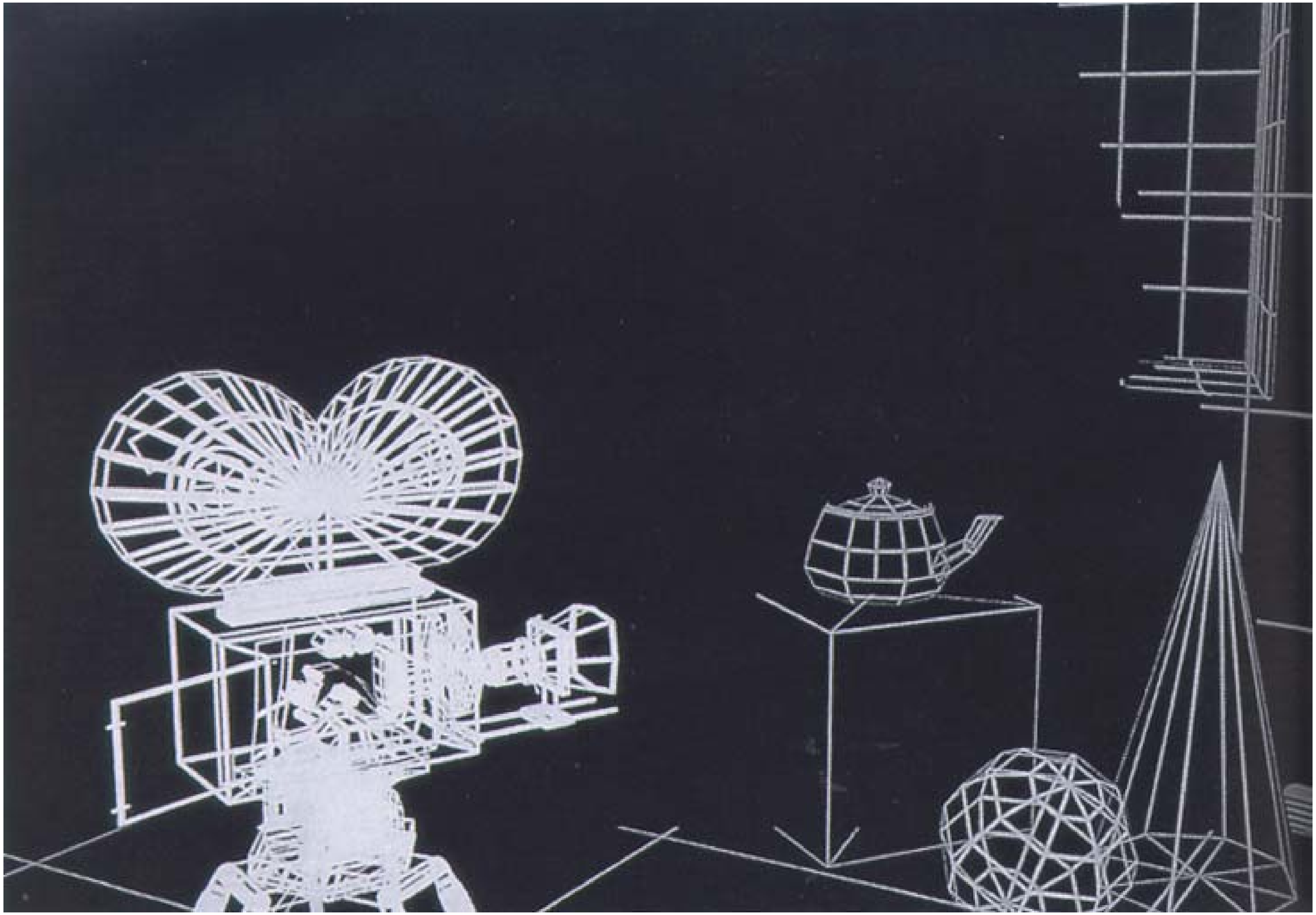
Polygonal Model Generated from Spline Patches. Perspective Projection

Example: Creating an Image from a Model



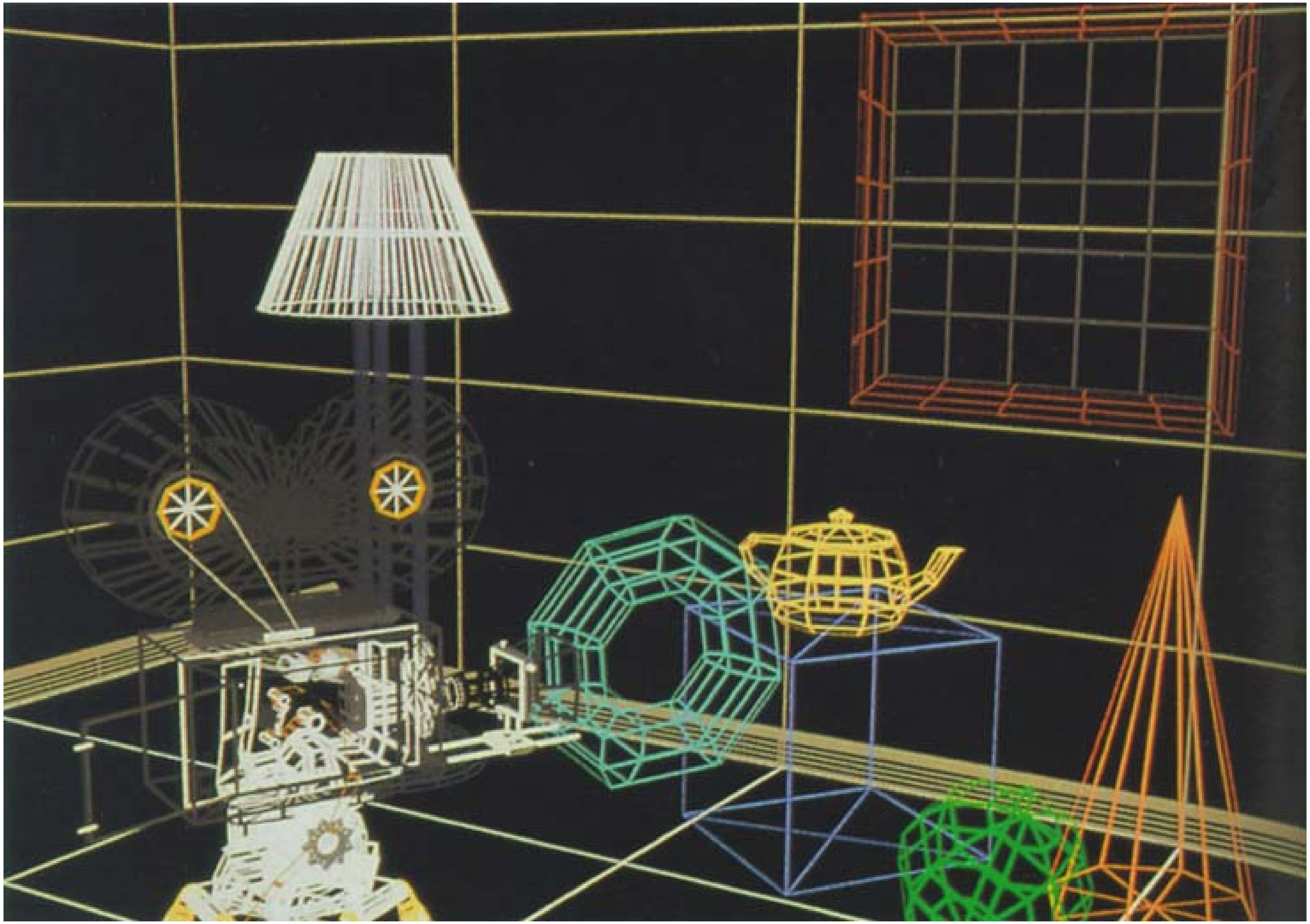
Depth Cueing

Example: Creating an Image from a Model



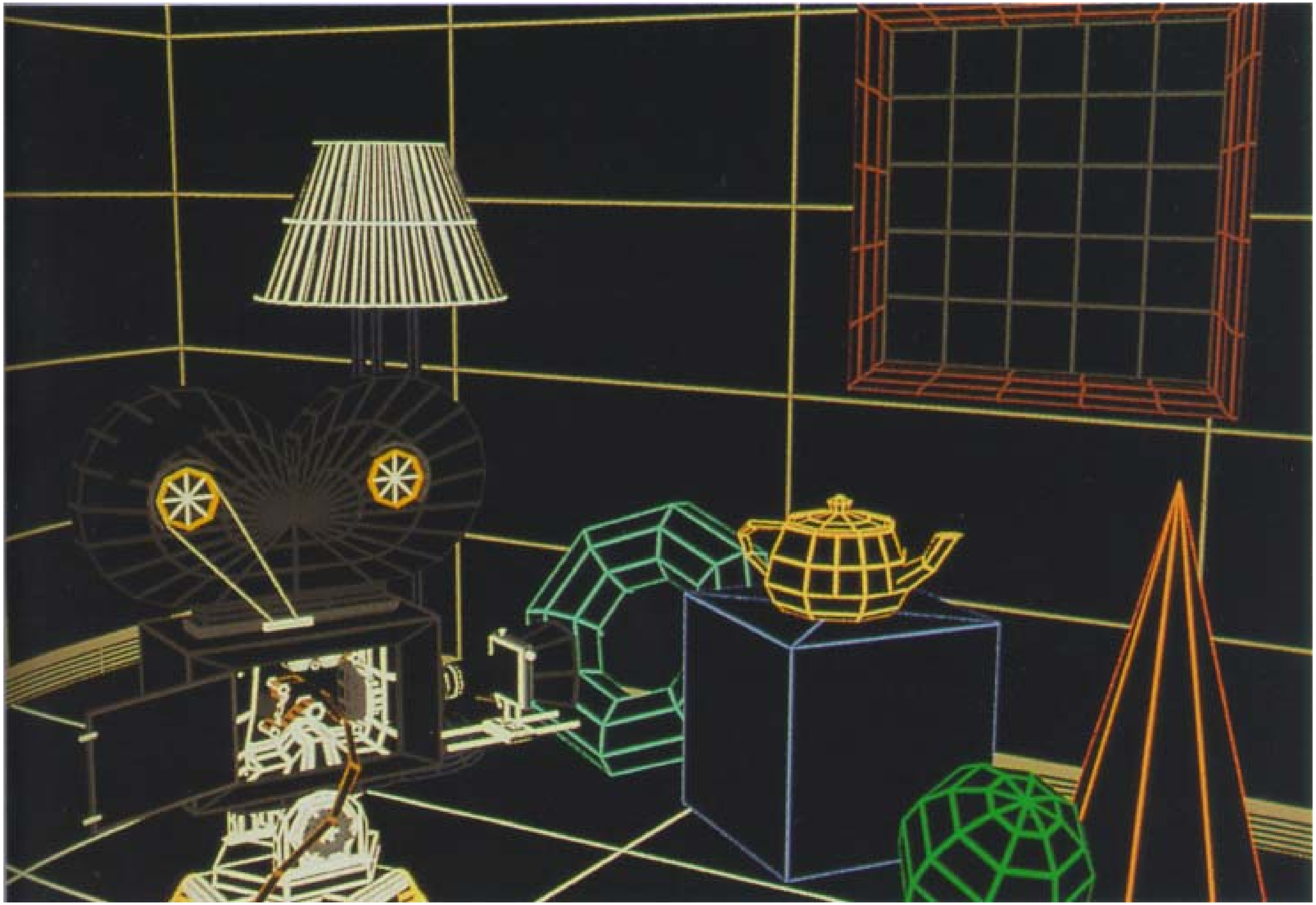
Depth Clipping

Example: Creating an Image from a Model



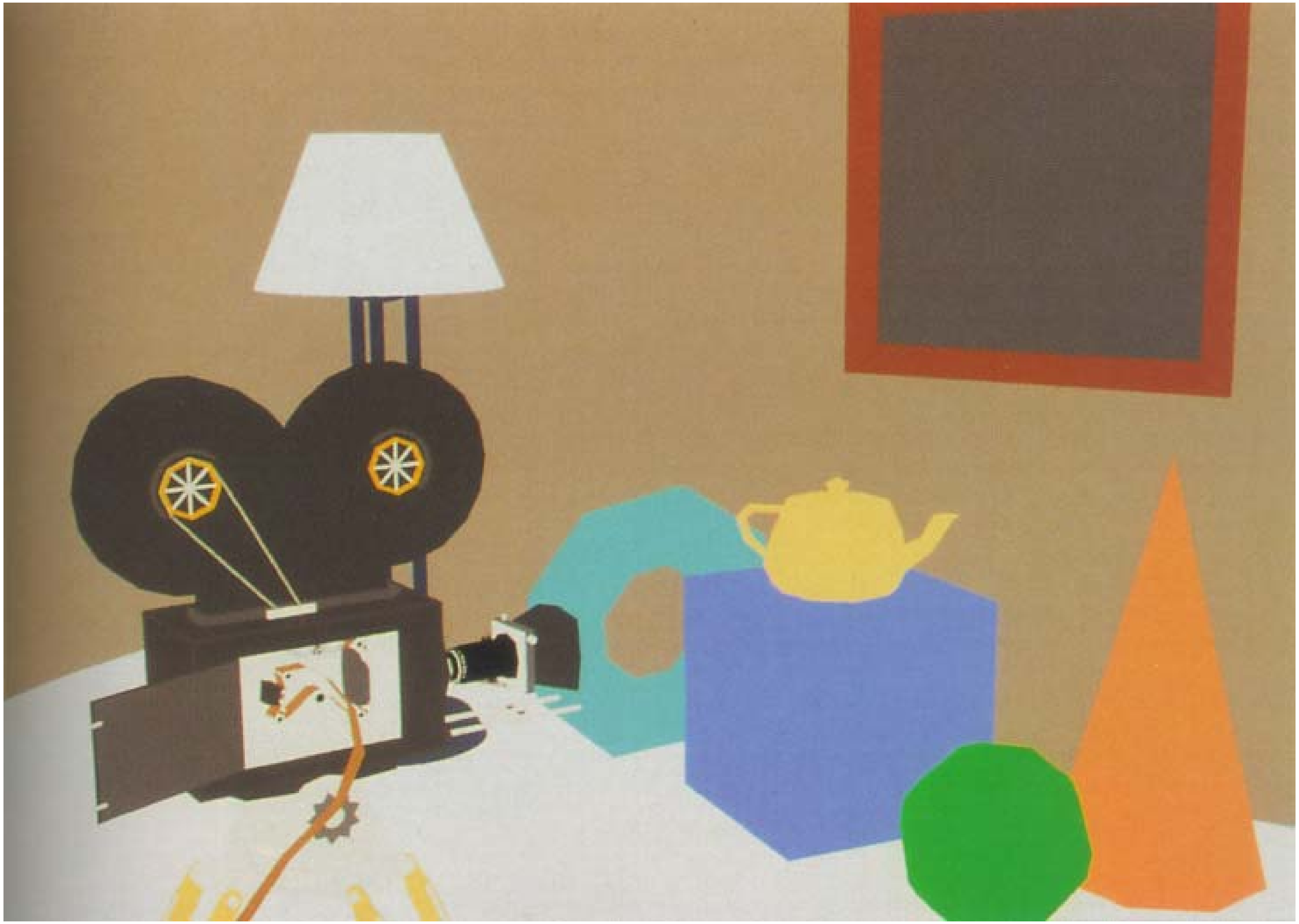
Colored Vectors

Example: Creating an Image from a Model



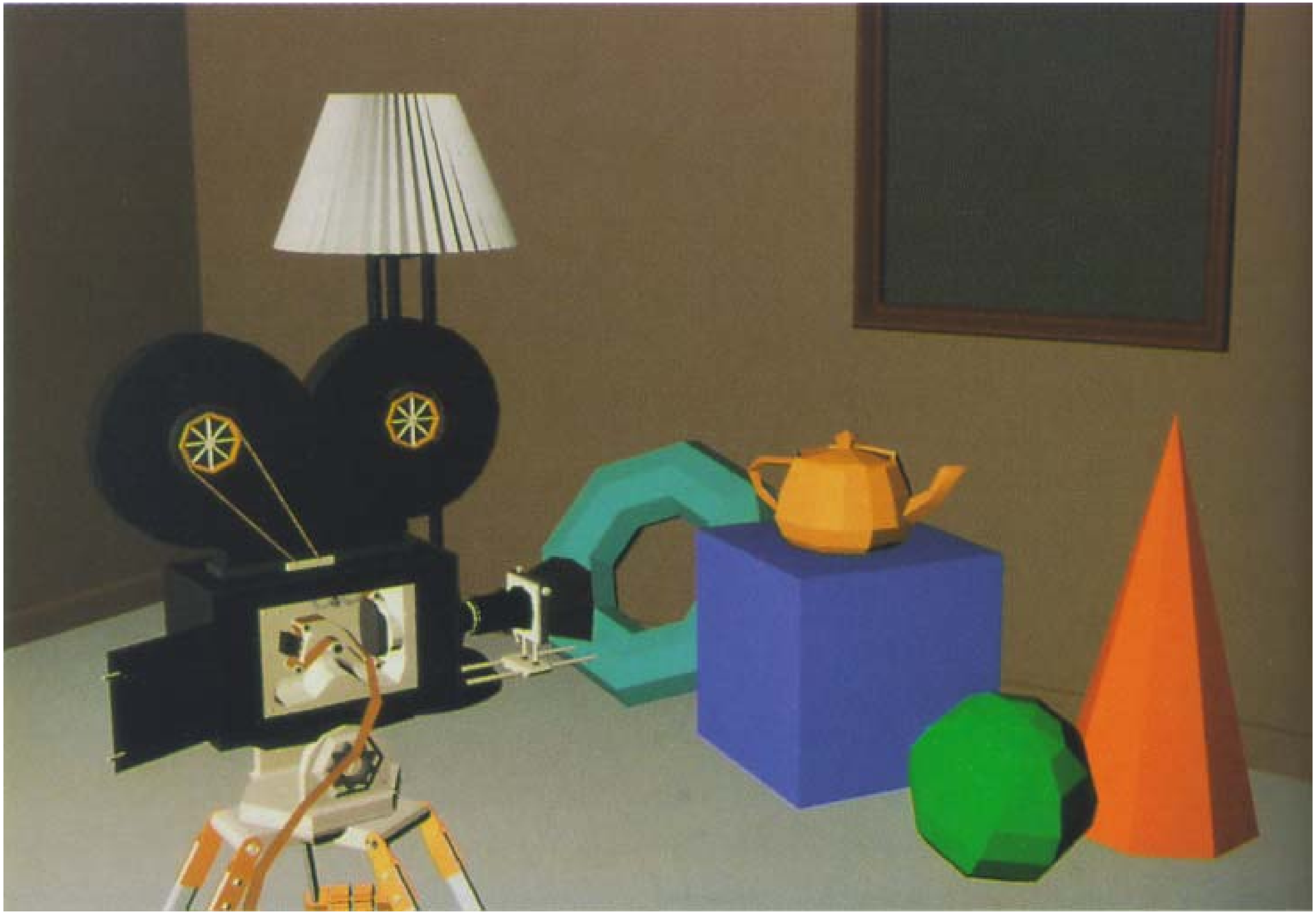
Visible Line Determination

Example: Creating an Image from a Model



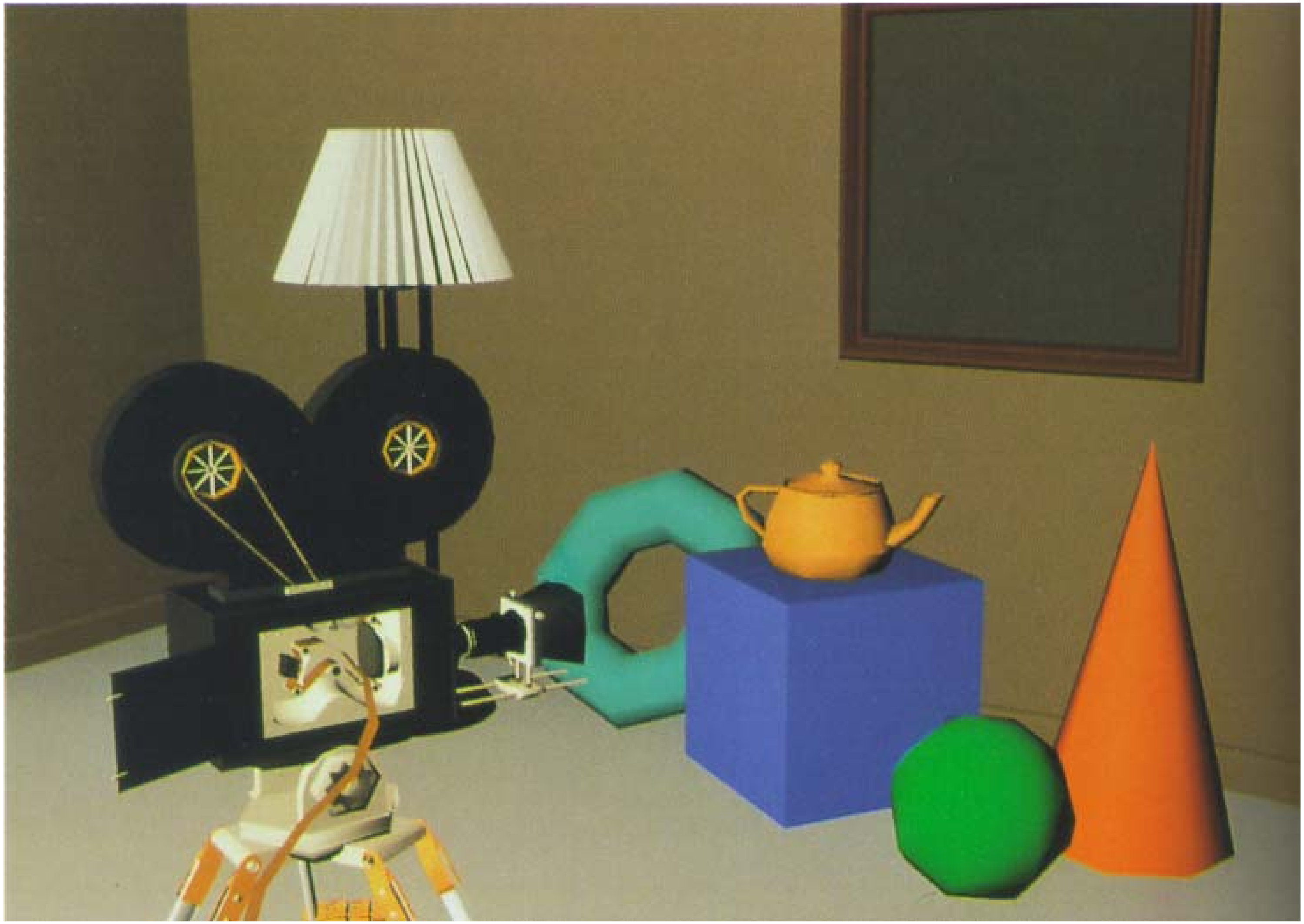
Visible Surface Determination with Ambient Illumination

Example: Creating an Image from a Model



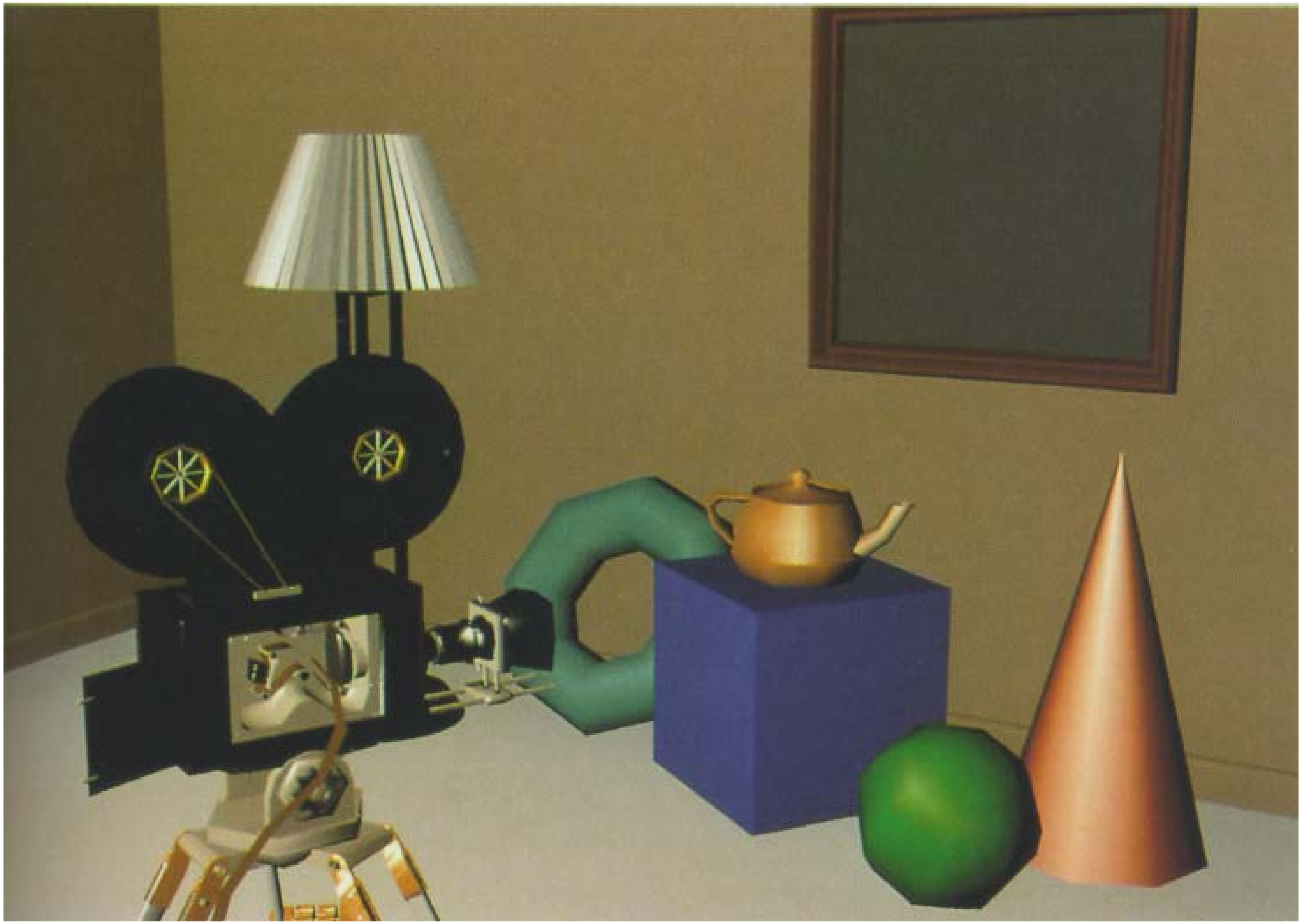
Individually Shaded Polygon with Diffuse Reflection

Example: Creating an Image from a Model



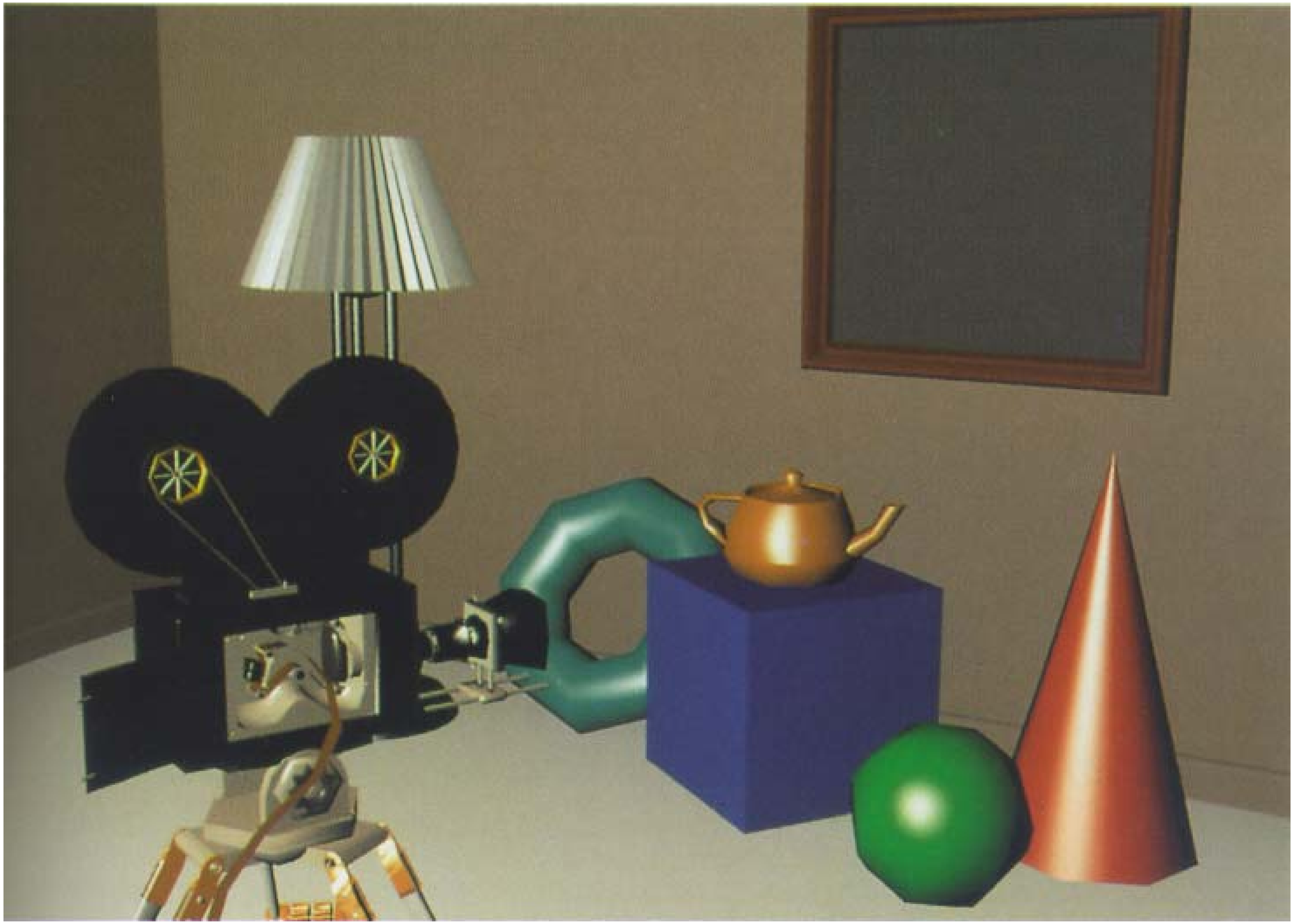
Gouraud Shaded Polygon with Diffuse Reflection

Example: Creating an Image from a Model



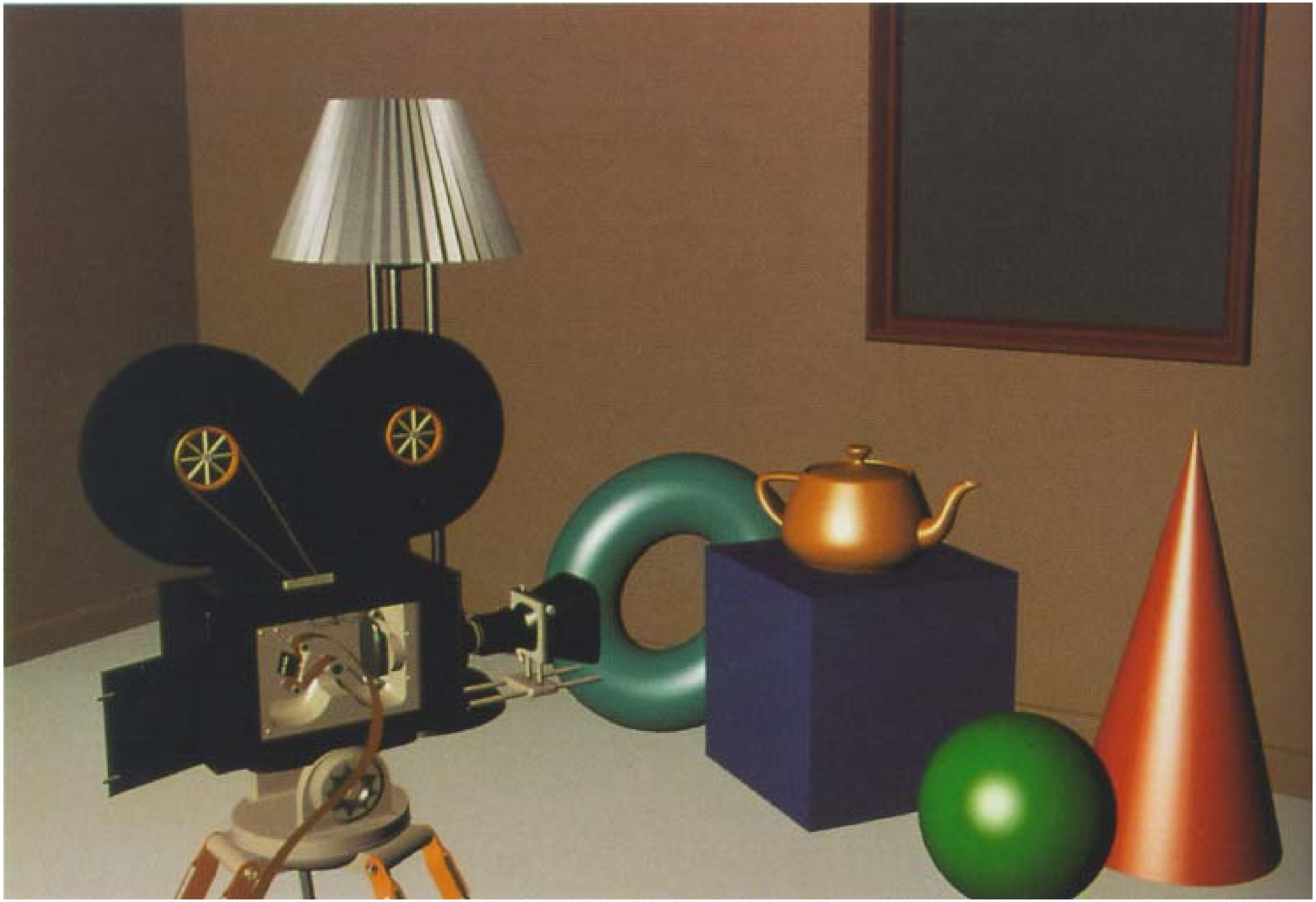
Gouraud Shaded Polygon with Specular Reflection

Example: Creating an Image from a Model



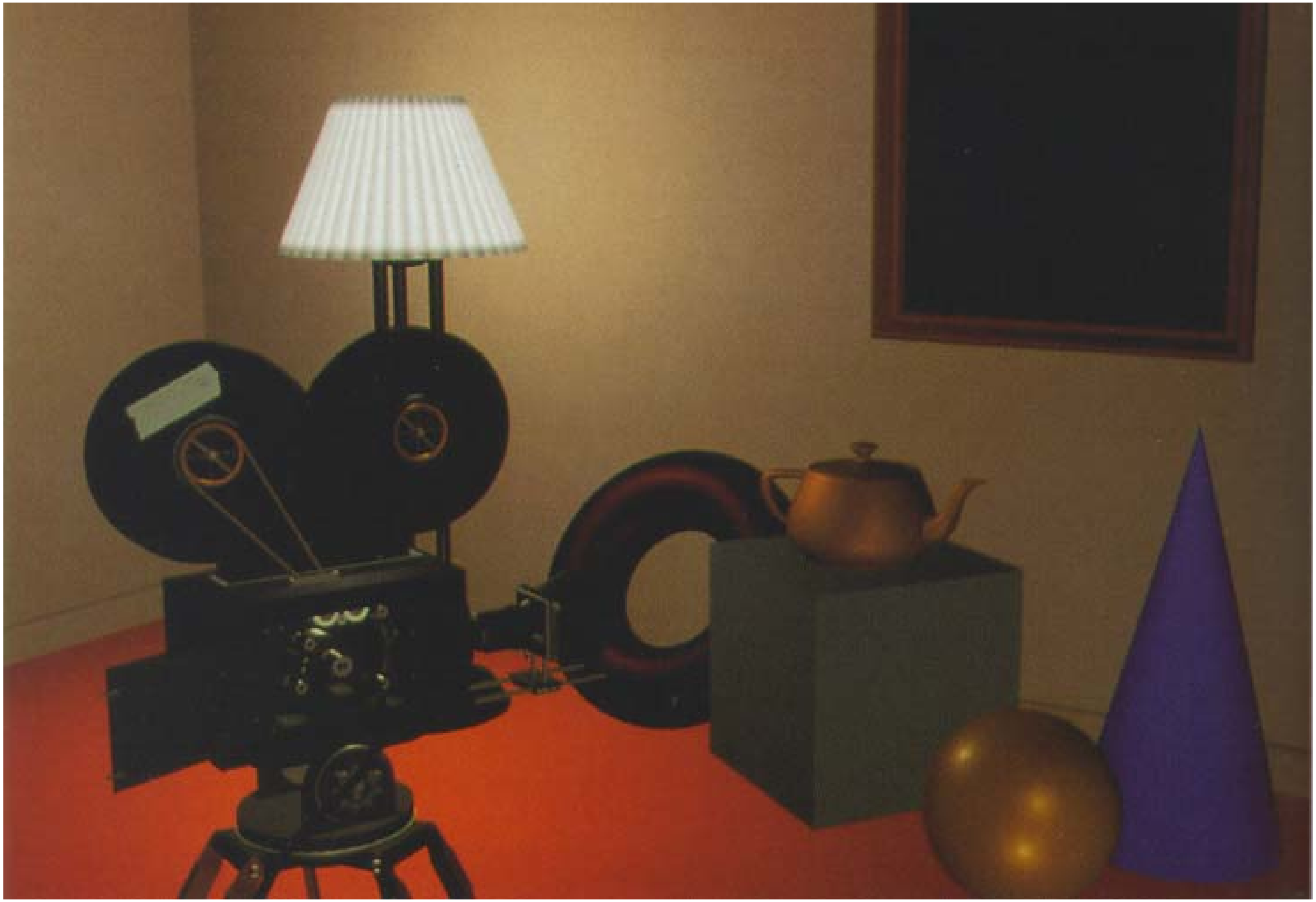
Phong Shaded Polygon with Specular Reflection

Example: Creating an Image from a Model



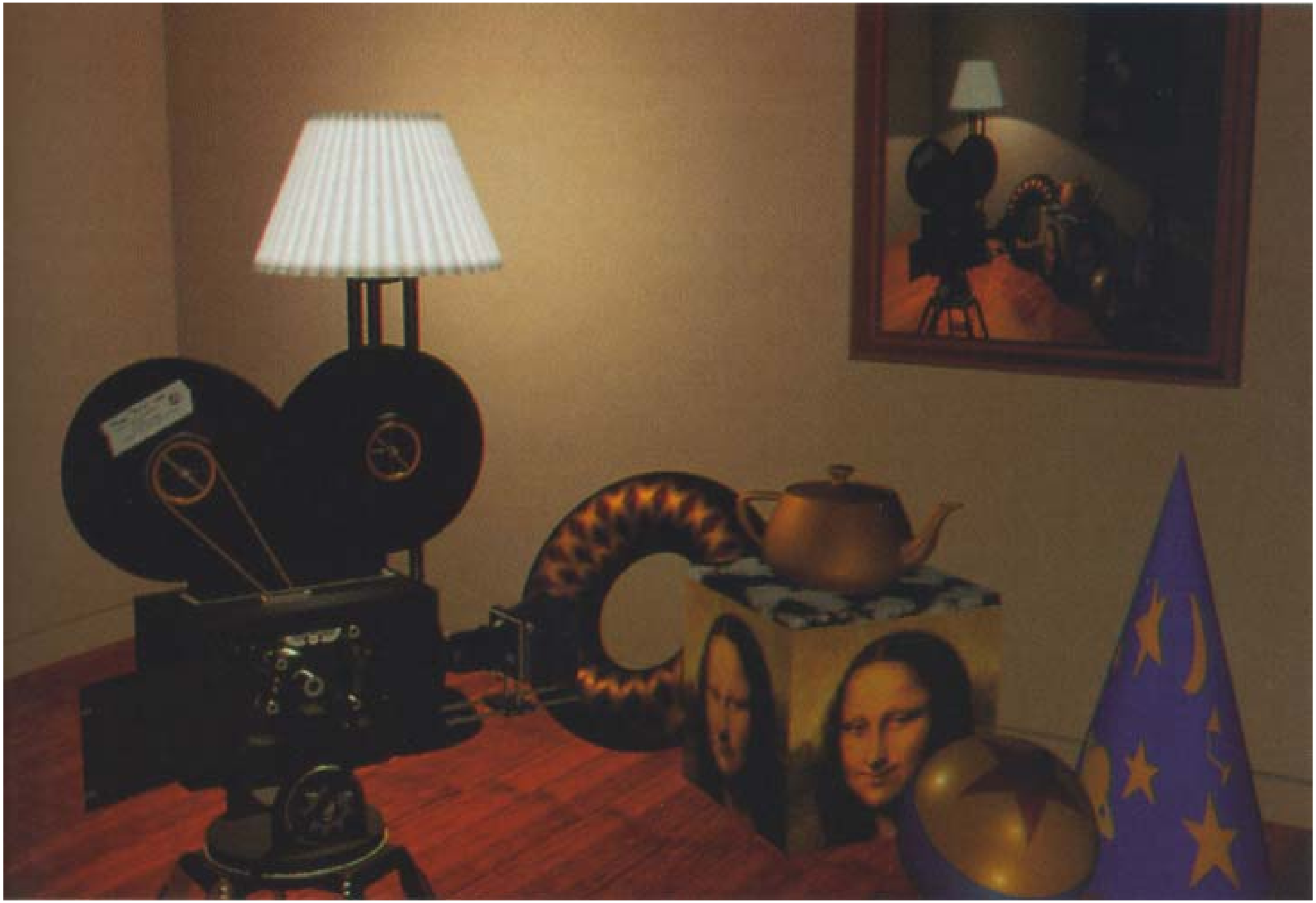
Curved Surfaces with Specular Reflection

Example: Creating an Image from a Model



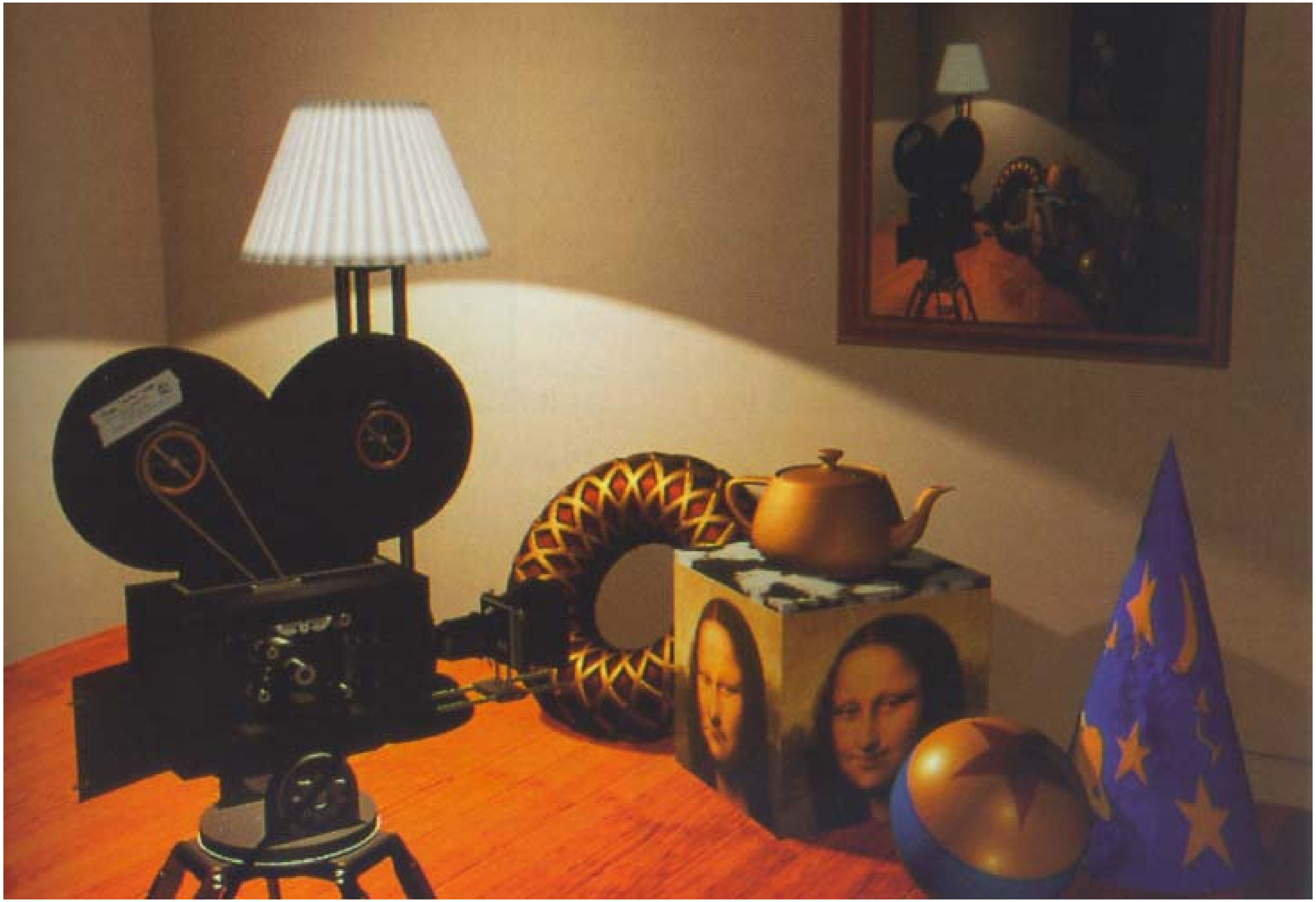
Multiple Lights

Example: Creating an Image from a Model



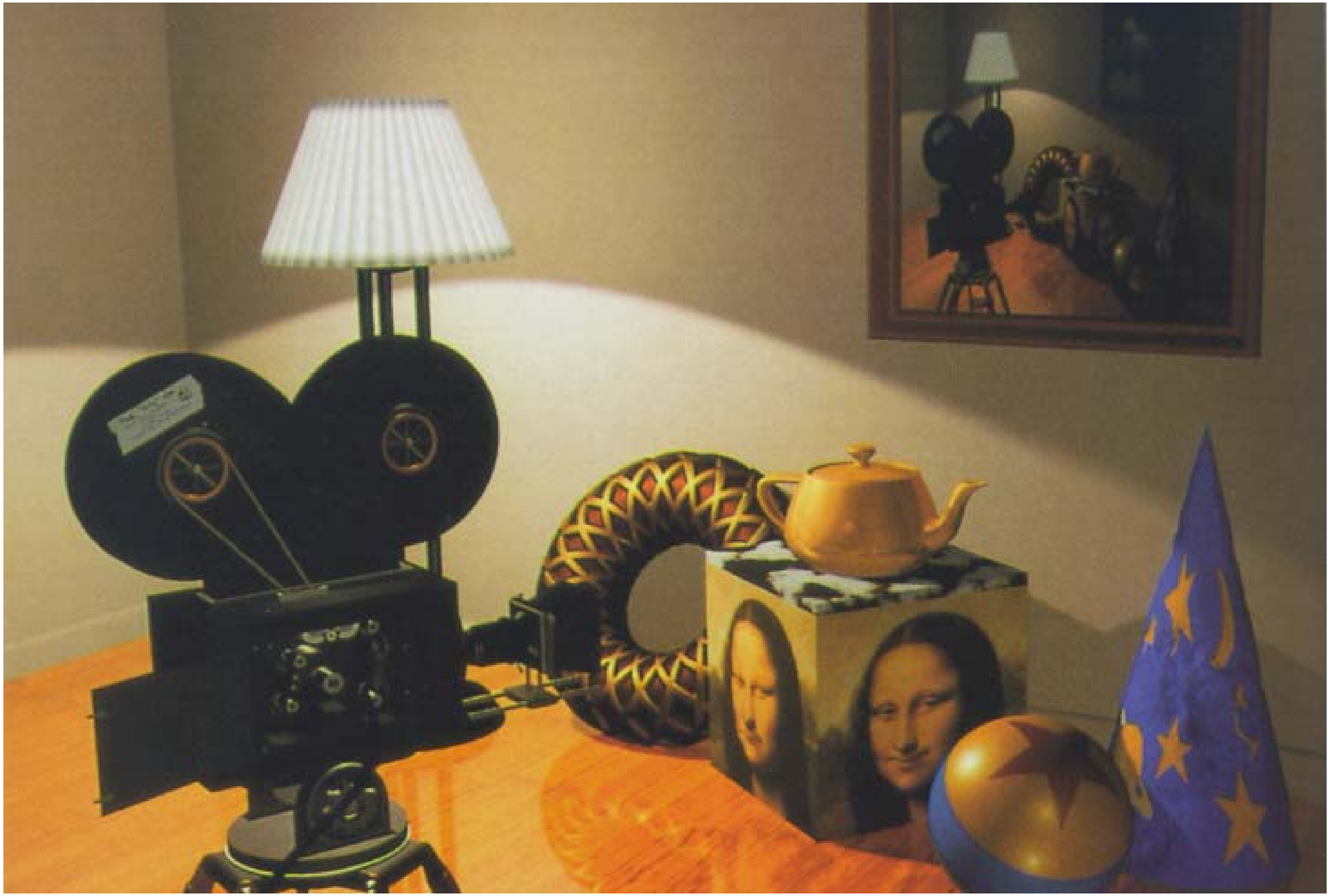
Texture Mapping

Example: Creating an Image from a Model



Shadows

Example: Creating an Image from a Model

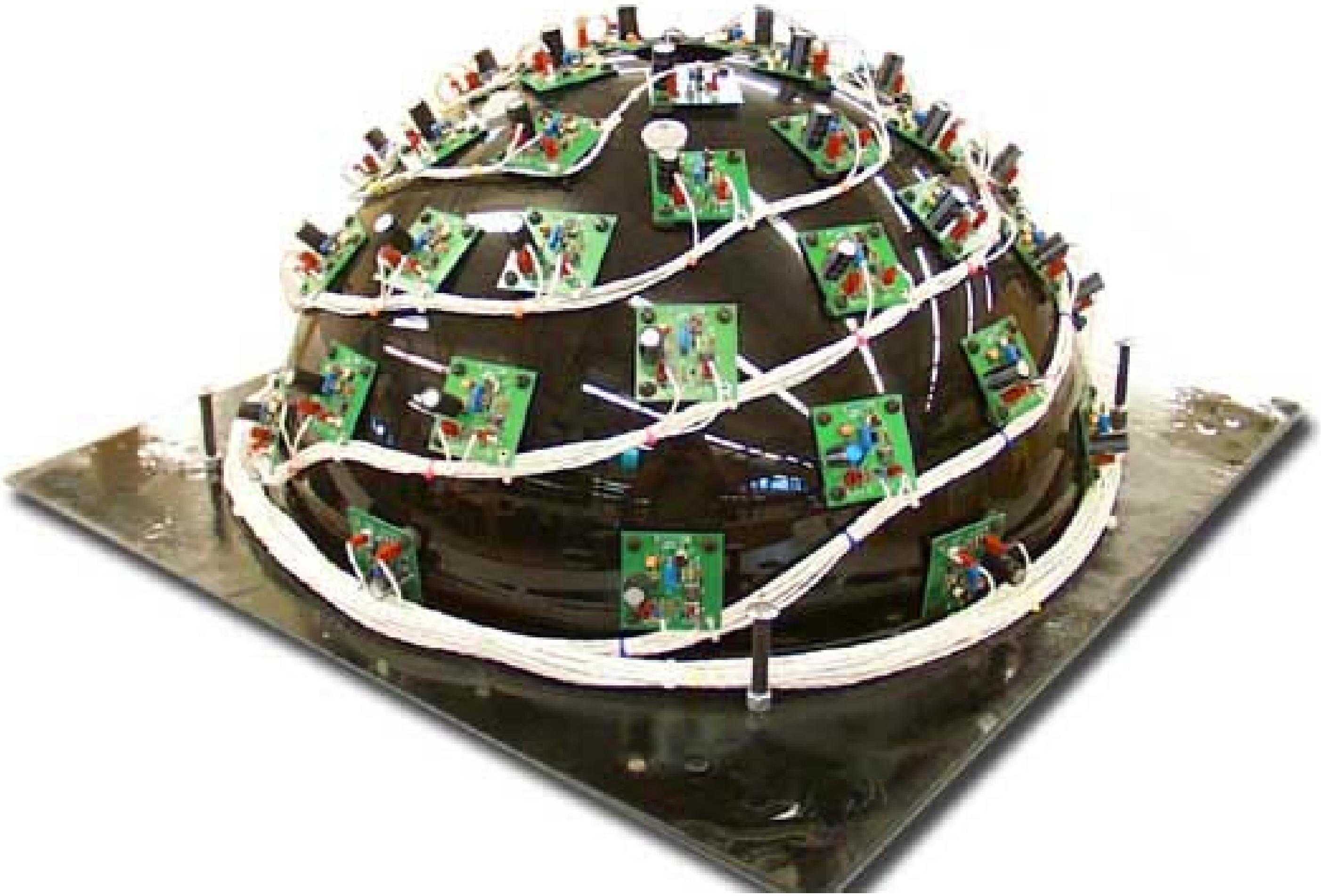


Reflection Mapping

Example: Polynomial Texture Maps

From:

http://www.hpl.hp.com/research/ptm/



Example: Polynomial Texture Maps

From:

http://www.hpl.hp.com/research/ptm/