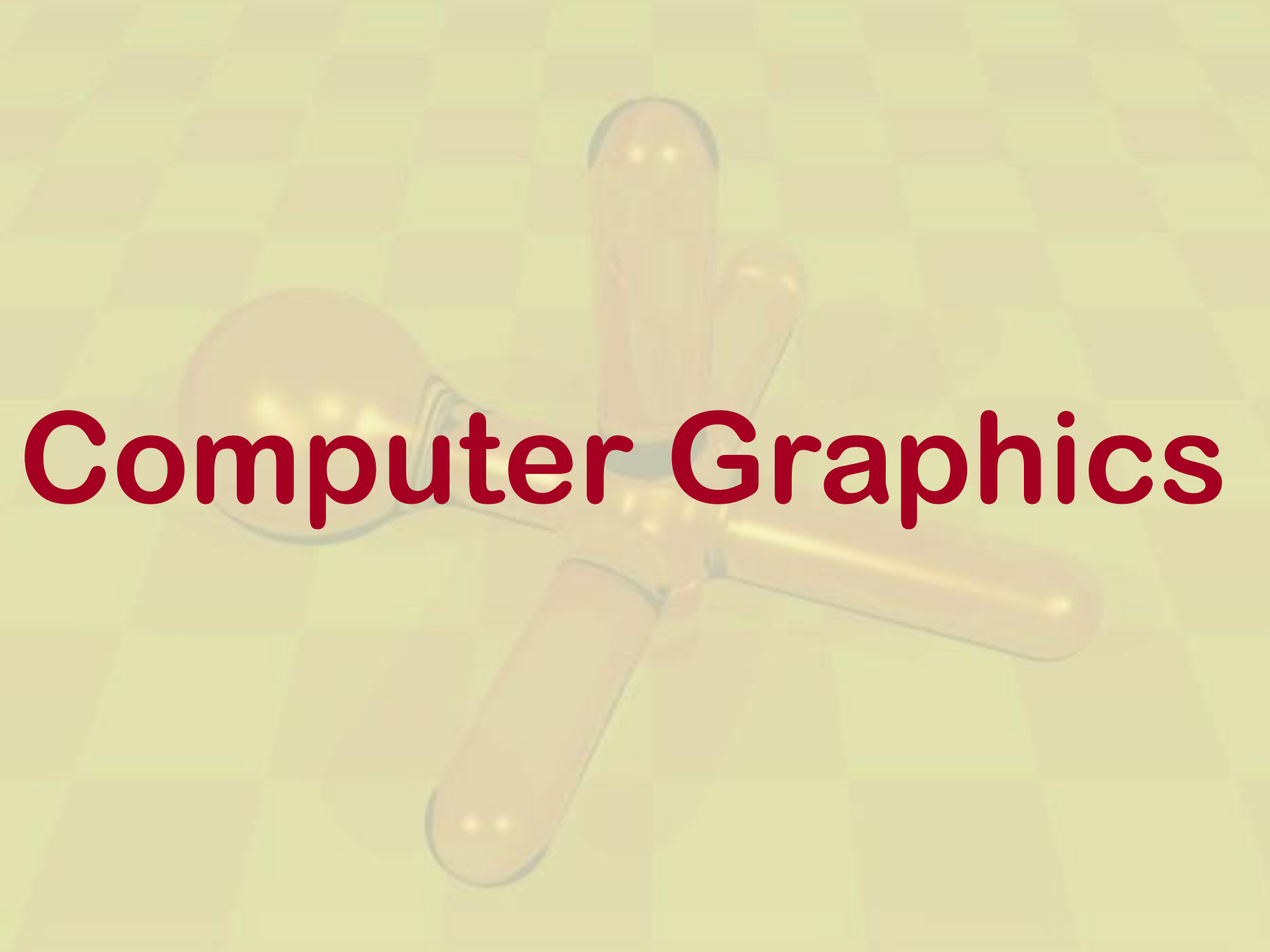

Computer Graphics

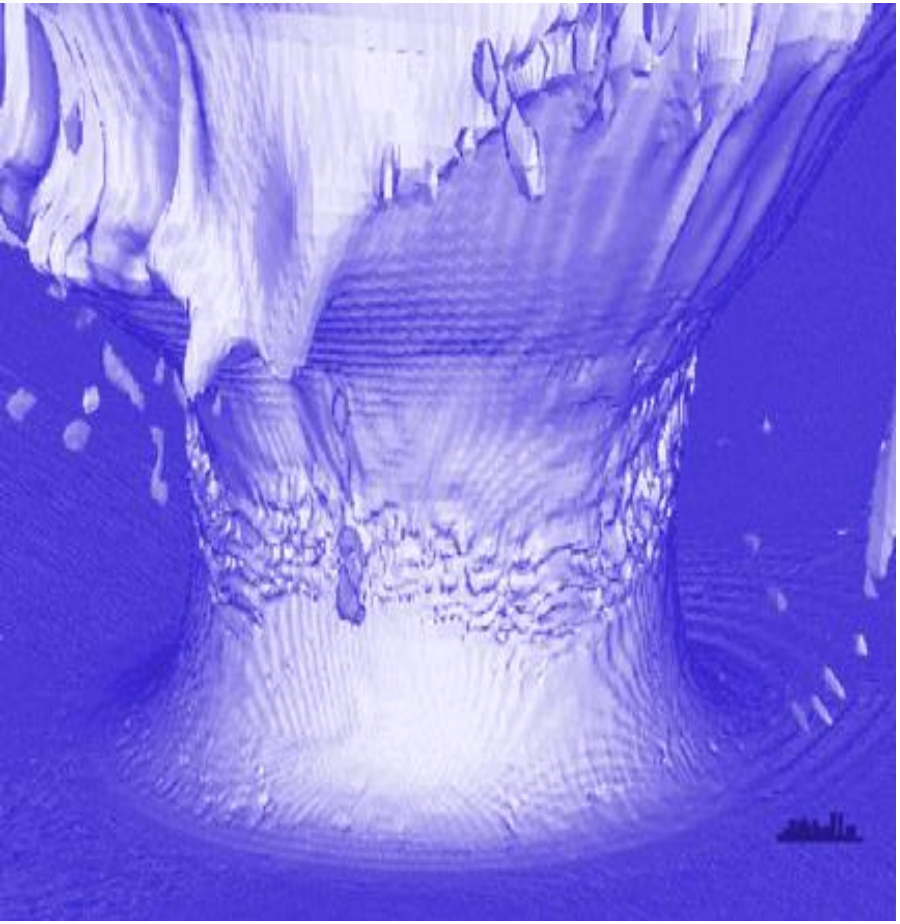
Aspects of Graphics

- Design vs. Programming
- Interactive vs. Photorealistic
- 2D vs. 3D
- Graphics vs. image processing vs. user interfaces

Graphics Software

- Photoshop, Illustrator, etc.
- 3D Modeling (CAD, animation)
- Rendering (ray tracing, radiosity)
- Animation tools
- Graphics programming APIs (OpenGL, DirectX)
- Scene graph libraries
- Game engines

Comet Simulation



COMET CRASH - Sandia supercomputer simulations of a one-kilometer comet entering Earth's atmosphere, approaching the ocean's surface, and impacting the ocean, deforming the ocean floor and creating a giant high-pressure steam explosion rising into the stratosphere. The explosion ejects comet vapor and water vapor into ballistic trajectories that spread around the globe. The New York City skyline is shown for scale.


Renzo Del Fabbro - Editor - Povray 3

OpenGL

- OpenGL: (Open Graphics Library)
- a widely-used, open API for 3D graphics
  - Old, originally from Silicon Graphics (SGI)
  - Low-level, procedural (vs. scene graph retained mode)
  - Designed for speed, control over hardware
  - Need hardware support for top performance
  - Widely used for CAD, VR, visualization, flight simulators
  - Managed by non-profit “Khronos Group” consortium
- Support
  - All major graphics cards, platforms have support
  - Mobile devices (iOS, Android) use an embedded version
  - HTML5 has experimental WebGL support
  - Bindings for JavaScript, Java, C#, Perl, Python, Ruby, Scheme, Visual Basic, Ada, …

Graphics Only

- OpenGL does not support windowing, interaction, UI, etc
- It must be used with another windowing system/library such as
  - MS Windows—various
  - Cocoa
  - X11
  - Qt
  - GLUT, GLFW
  - HTML5 JavaScript?

Refraction using vertex shaders

OpenGL vs. Direct3D

- Direct3D:
  - MS only
  - Used more for games
  - Latest versions are good
- OpenGL
  - Used more for professional applications
  - Not much development until a few years ago
  - Mobile gaming mostly on OpenGL ES (Embedded System)
  - Unreal, Unity, other game engines on OpenGL ES

WebGL

- **WebGL** (*Web Graphics Library*) is a JavaScript API for rendering interactive 3D graphics and 2D graphics within any compatible web browser without the use of plug-ins.
- OpenGL 2.0 ES in your Web browser, no plugins needed!
- Supported by all major browsers except Internet Explorer (Microsoft hates Web standards, OpenGL)
- Working group: Apple, Google, Mozilla, Opera (not Microsoft)

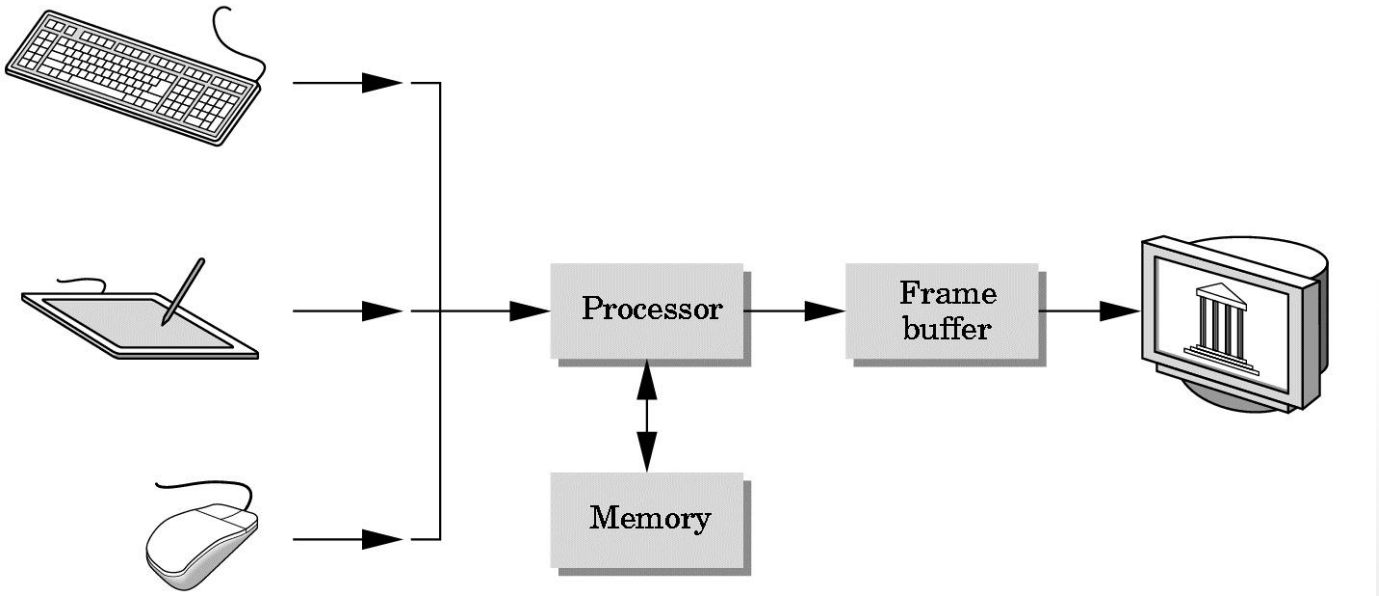
Other software

- **POV ray-tracer(** Persistence of Vision *Raytracer***)** is a ray tracing program which generates images from a text-based scene description, and is available for a variety of computer platforms.
- **ImageMagick** image manipulation library
- **3D Modeling**: Google's SketchUp or Blender
- HTML5 Canvas element for 2D graphics
  - The only cross-platform environment nowadays…

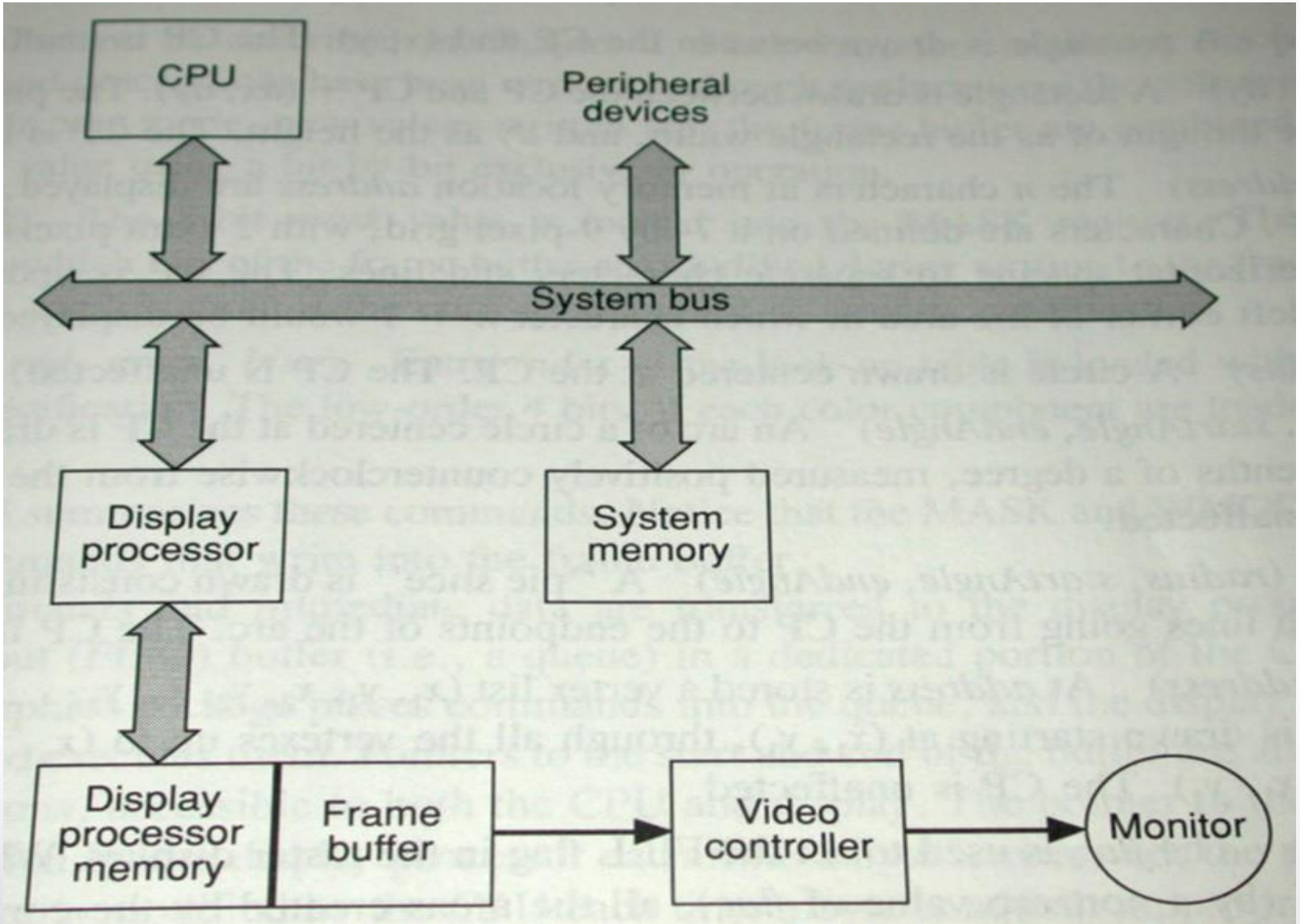
Chapter 1: Graphics Systems and Models

- A Graphics System
  - Processor
  - Memory
  - Frame Buffer

- Display
- Input Devices
- Output Devices

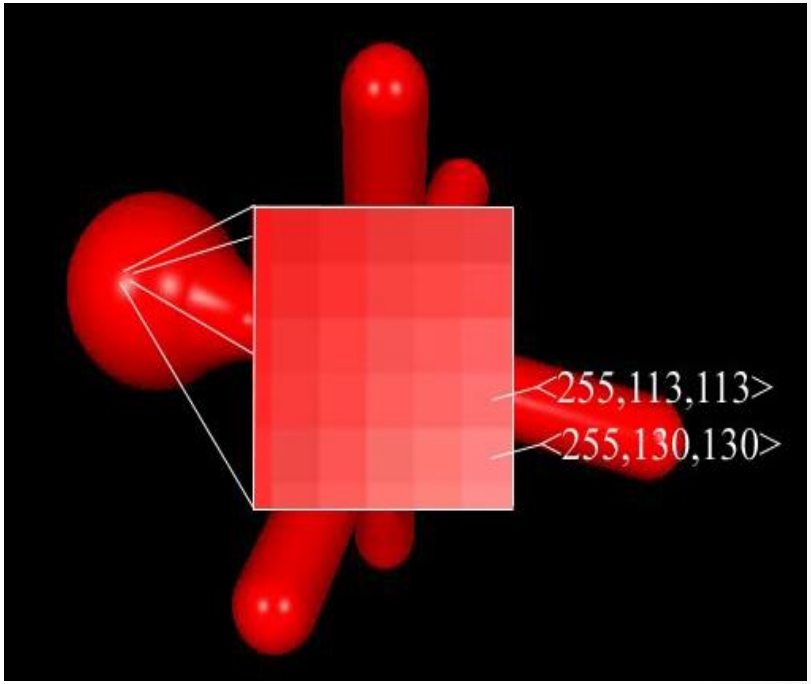


Graphics Architecture



Images

- Array of pixels
- Red, Green, Blue
- May also have an *alpha* value (opacity)



Pixels and the Frame Buffer

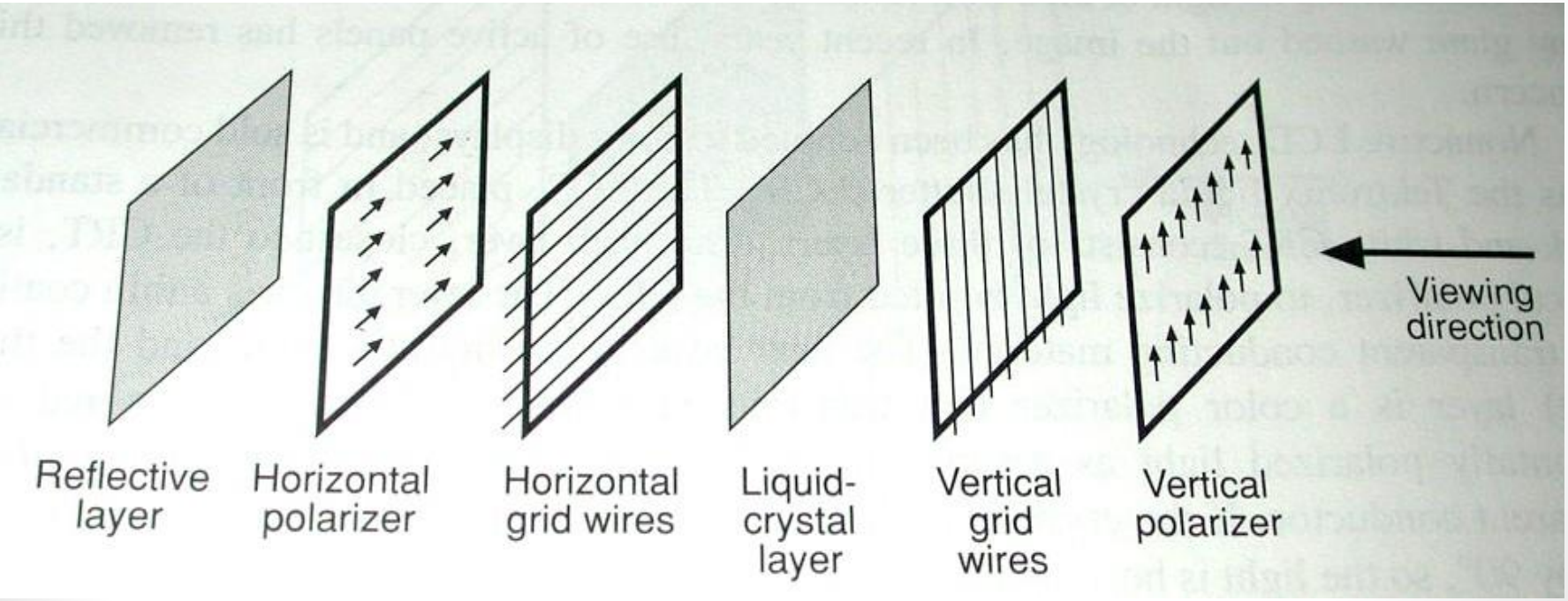
- Pixels:
  - picture elements
  - 3 values: RGB, 0-255 or 0-65536 or 0.0-1.0
  - 4 values: RGBA (Alpha = opacity)
- Frame buffer
  - Depth: bits per pixel
  - May have 24, 32, 64, or flexible depth

Display terms

- Scan line
- Resolution
- Horizontal and vertical re-trace
- Refresh, refresh rate
- Interlace
- NTSC, PAL, S-video, Composite
- HDTV

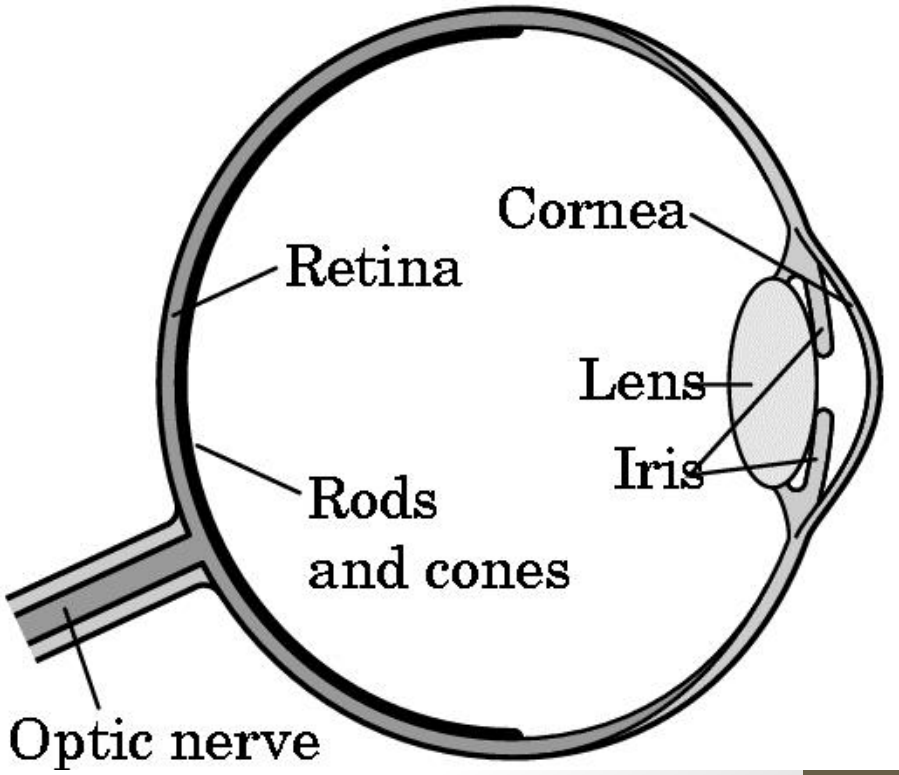
LCD Display

- An unpowered LCD layer changes polarization of light



The Human Visual System

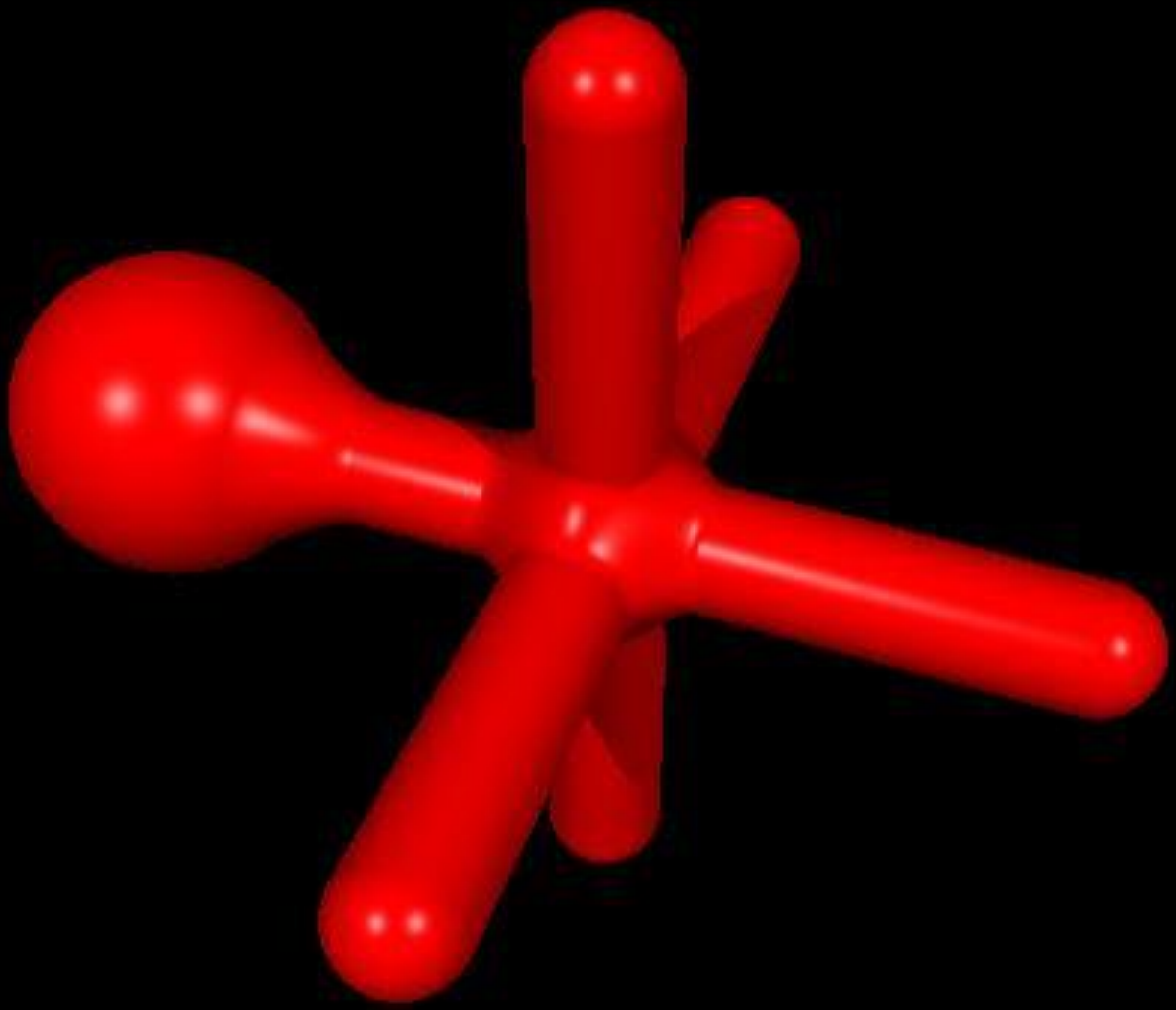
- Rods: night vision
- Cones: day vision
- Three types of cones, with different color sensitivity
- We model and render for its capabilities



Graphics Paradigms

- Modeling
- Rendering
  - Photo-realistic:
    - Ray tracing
    - Radiosity
  - Interactive:
    - Projection – camera model
    - Transformations, clipping
    - Shading
    - Texture mapping
    - Rasterization

Ray Tracing

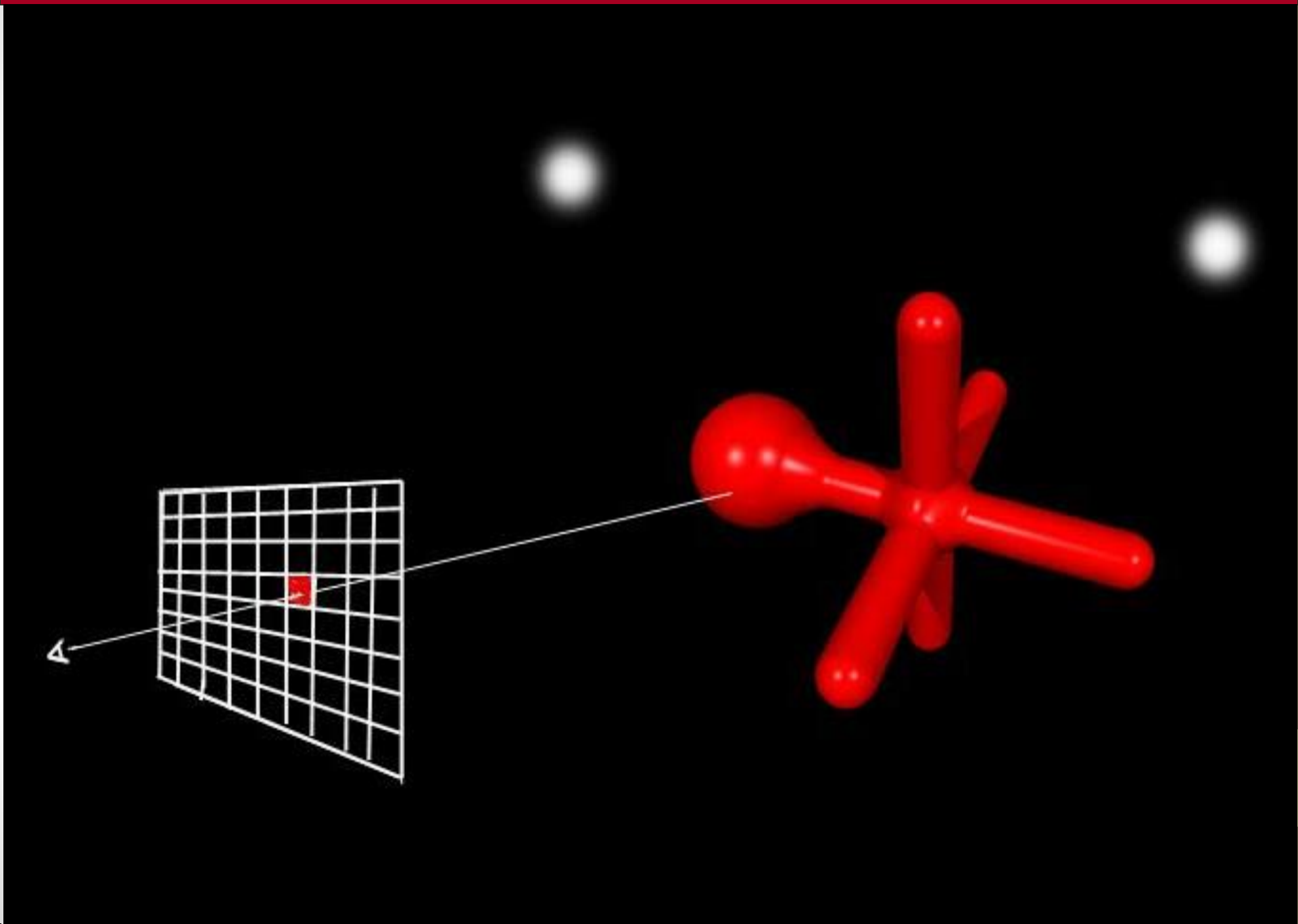


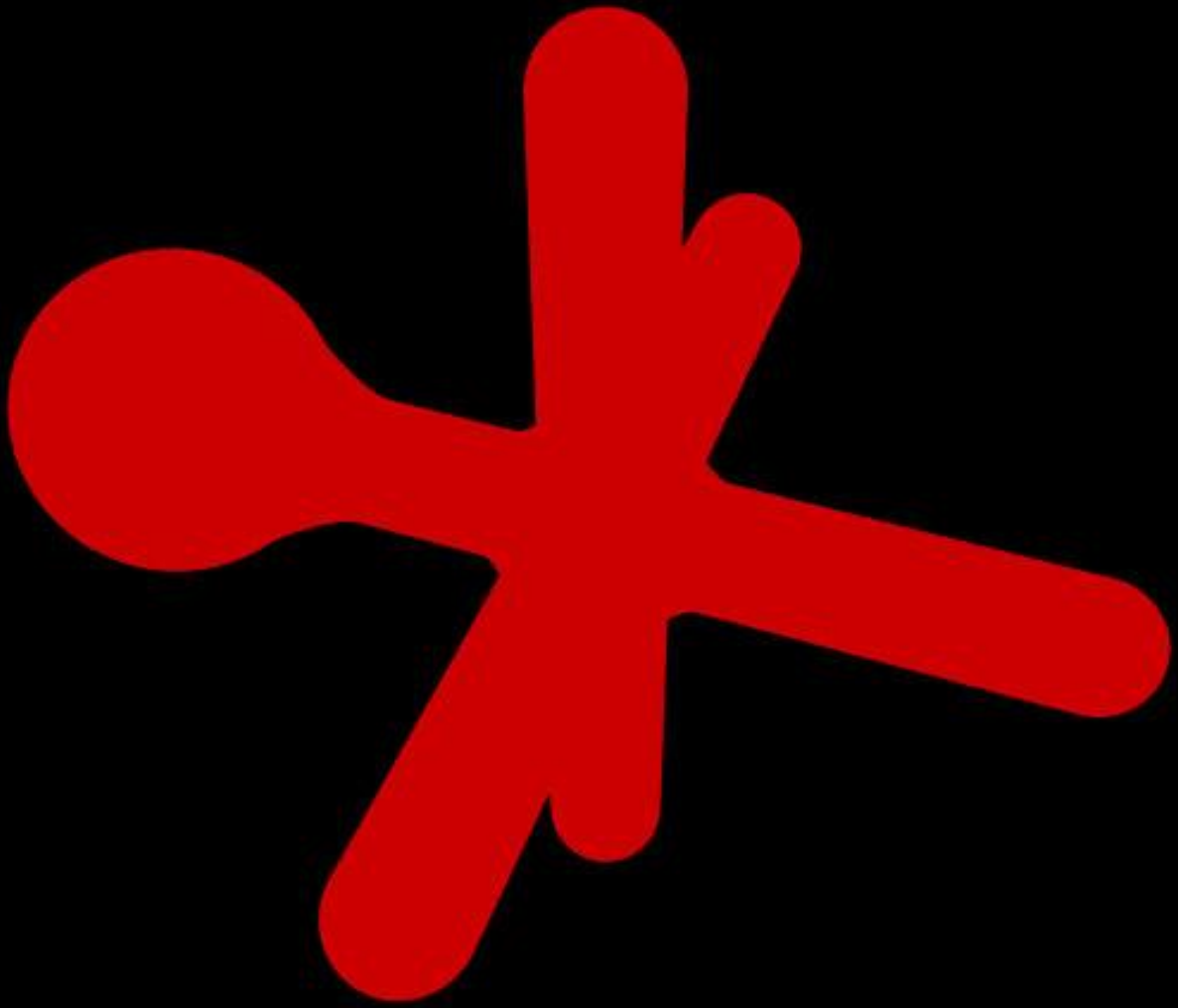
How does Ray-Tracing work?

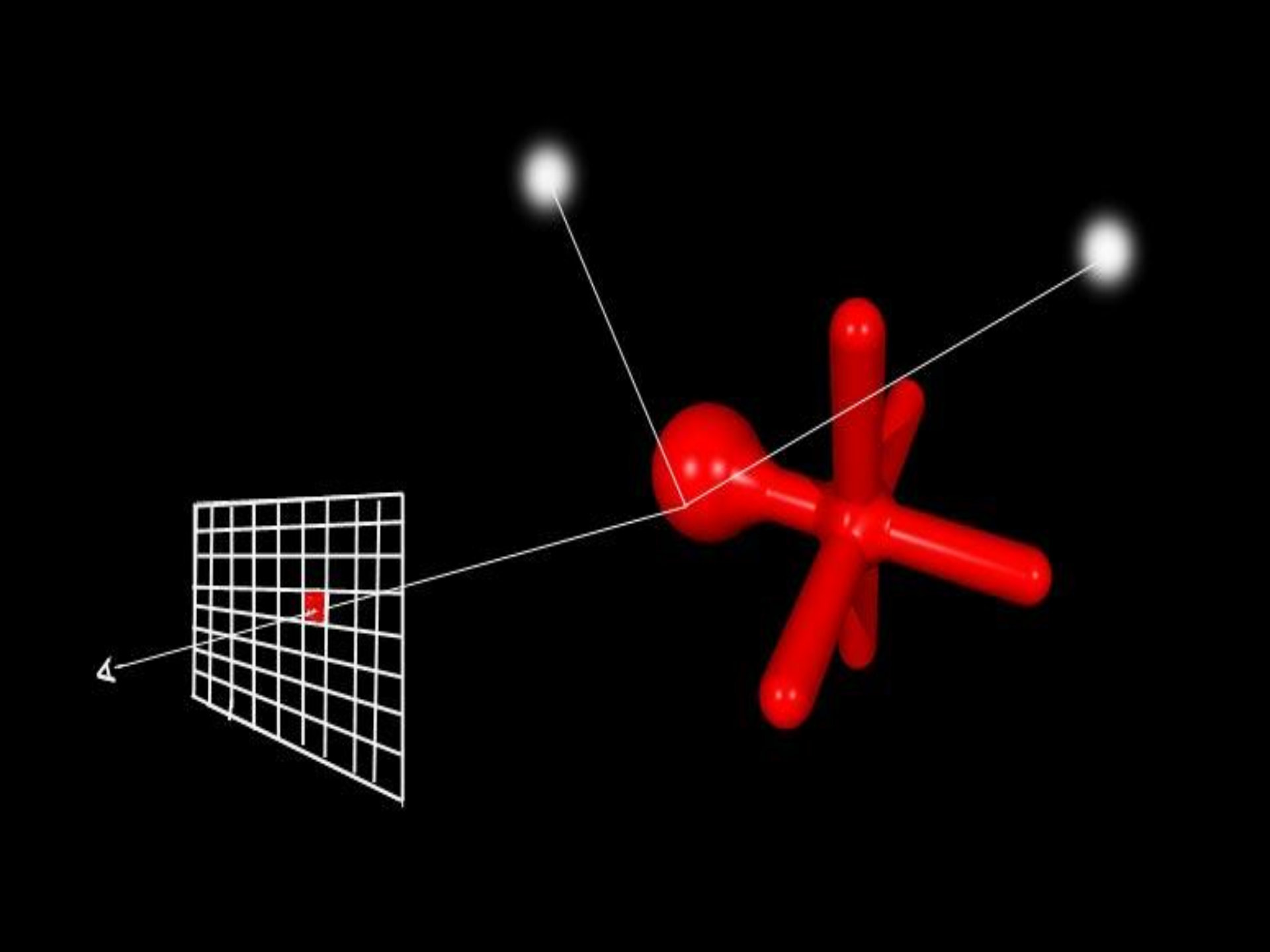
- Modeling
  - Build a 3D model of the world
    - Geometric primitives
    - Light sources
    - Material properties
  - Simulate the bouncing of light rays
    - Trace ray from eye through image pixel to see what it hits
    - From there, bounce ray in reflection direction, towards light source, etc.
    - Thus, model physics of emission, reflection, transmission, etc. (backwards)

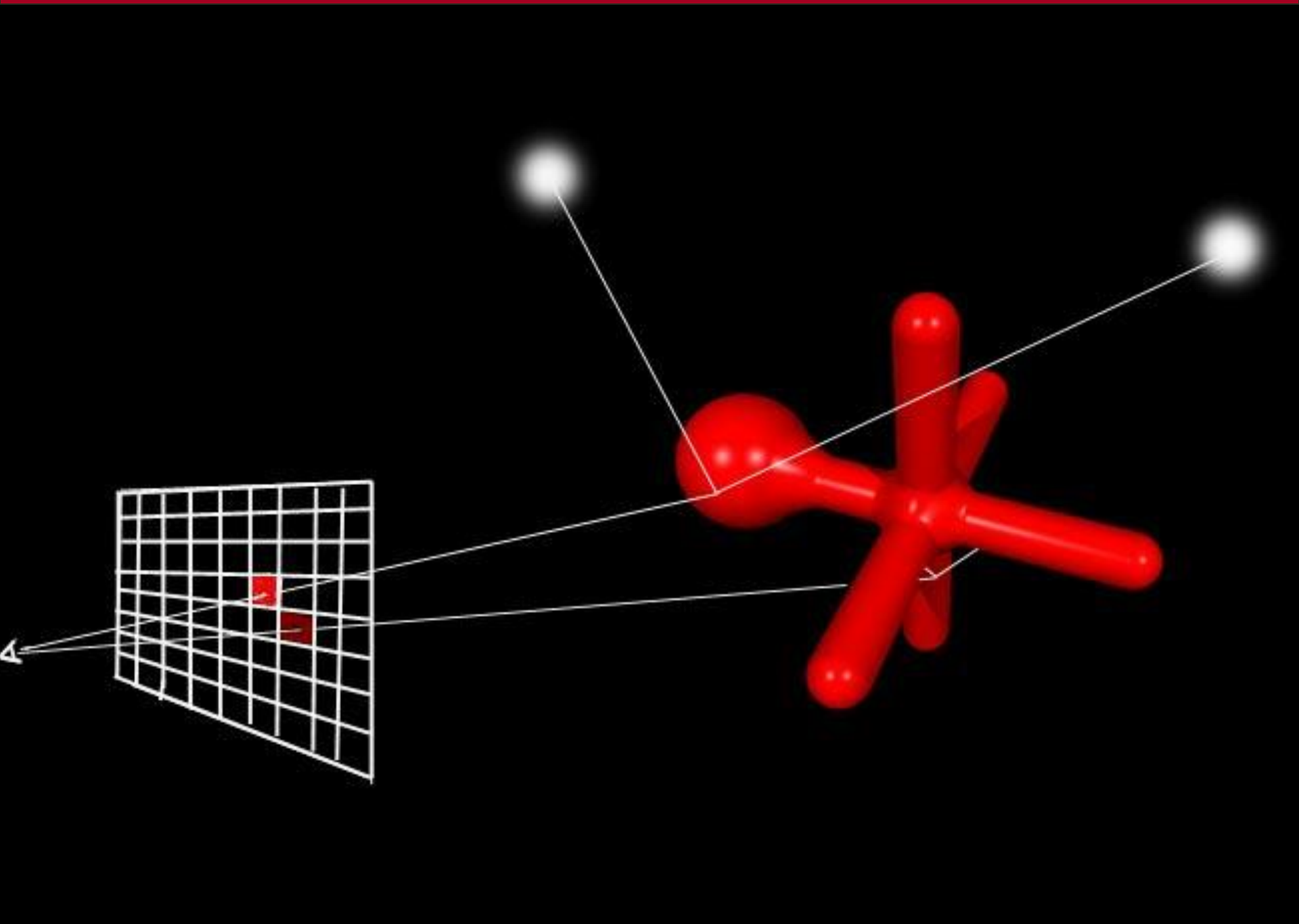
Modeling the World

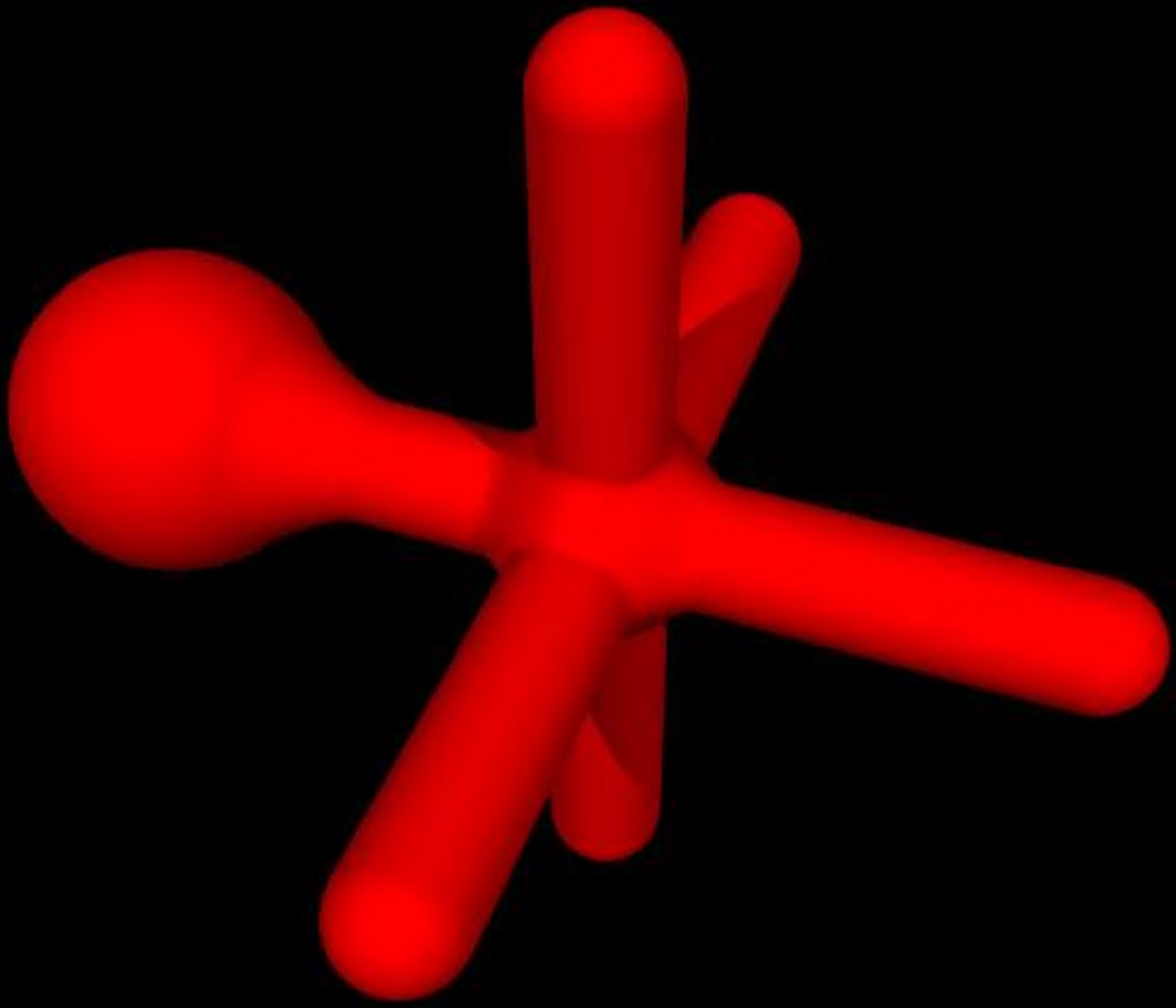
```
camera {
  location  <0, 5, -5>
  look_at   <0, 0, 0>
  angle 58
}

light_source { <-20, 30, -25>
  color red 0.6 green 0.6 blue 0.6 }

blob {
  threshold 0.5
  sphere   { <-2, 0, 0>, 1, 2 }
  cylinder { <-2, 0, 0>, <2, 0, 0>, 0.5, 1 }
  cylinder { <0, 0, -2>, <0, 0, 2>, 0.5, 1 }
  cylinder { <0, -2, 0>, <0, 2, 0>, 0.5, 1 }
  pigment  { color red 1 green 0 blue 0 }
  finish   { ambient 0.2 diffuse 0.8 phong 1 }
  rotate <0, 20, 0>
}
```

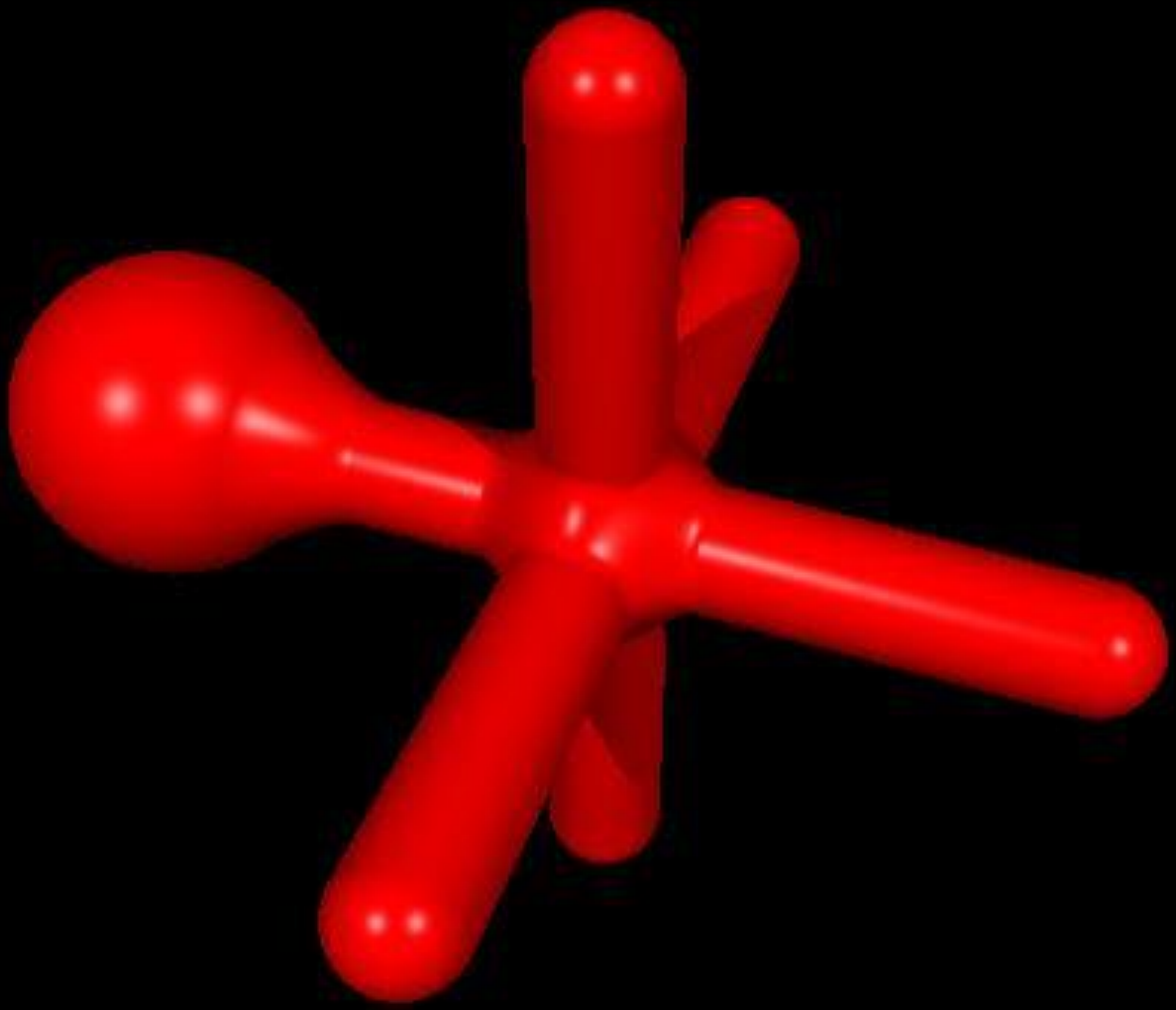


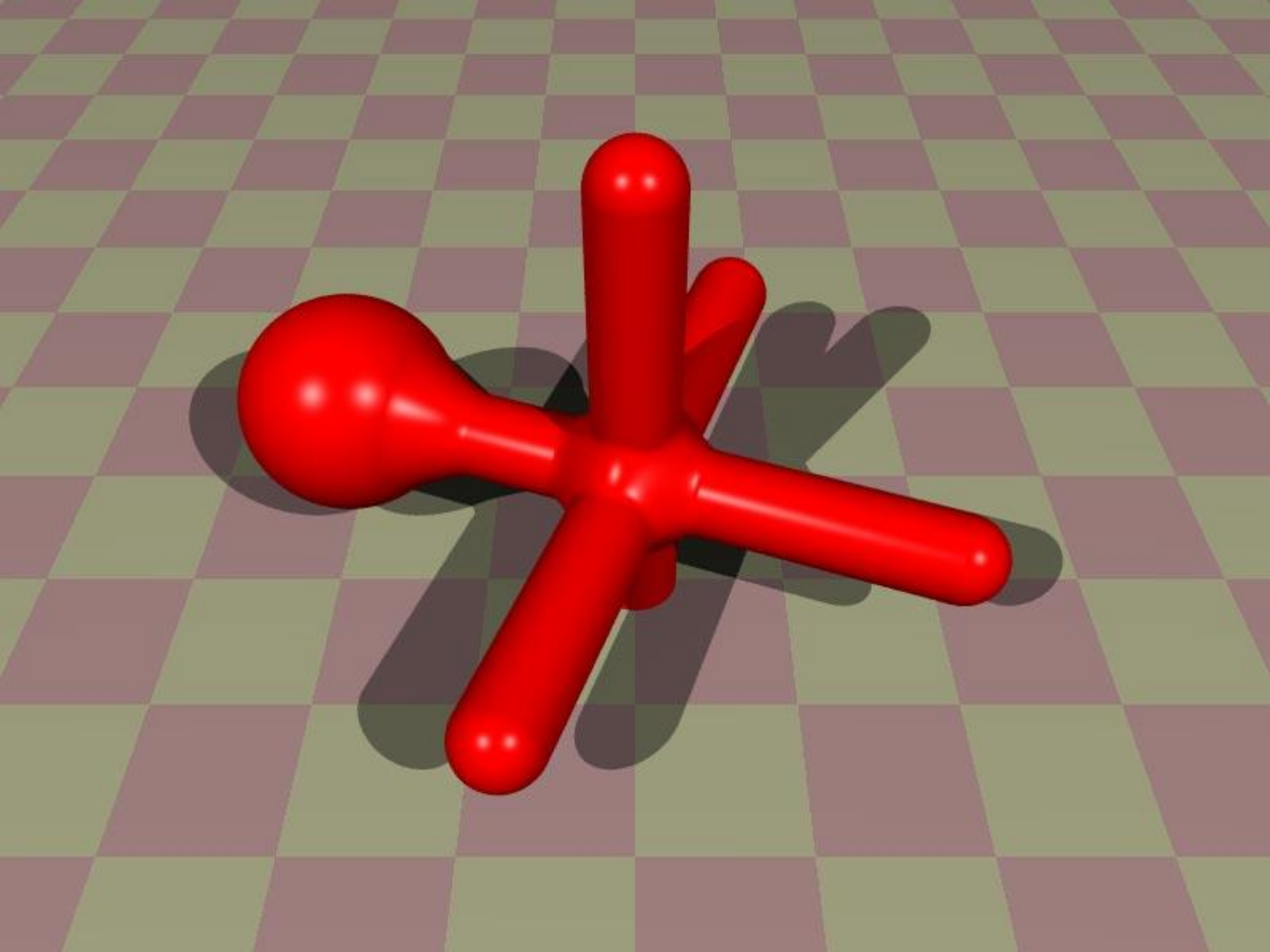


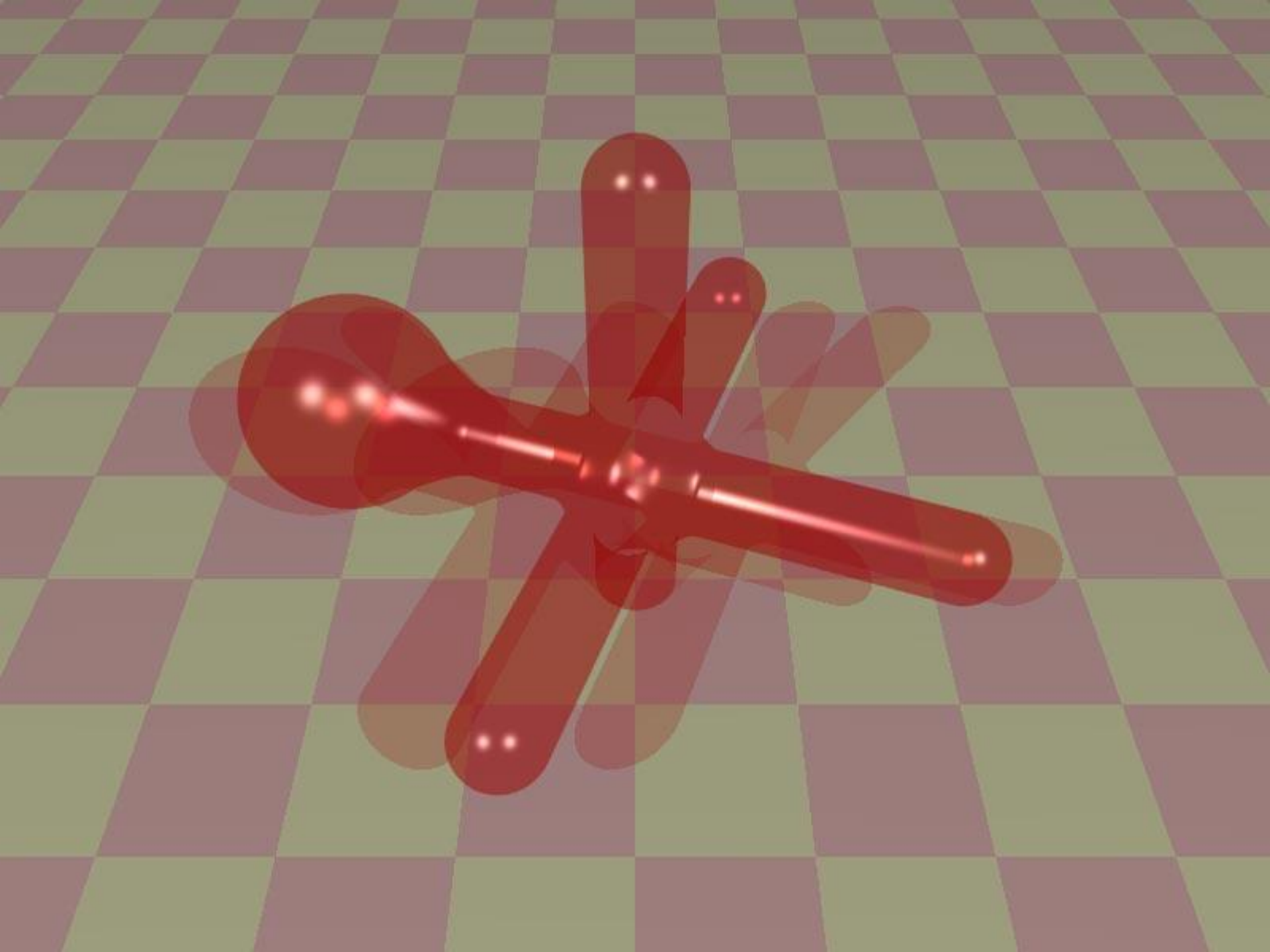


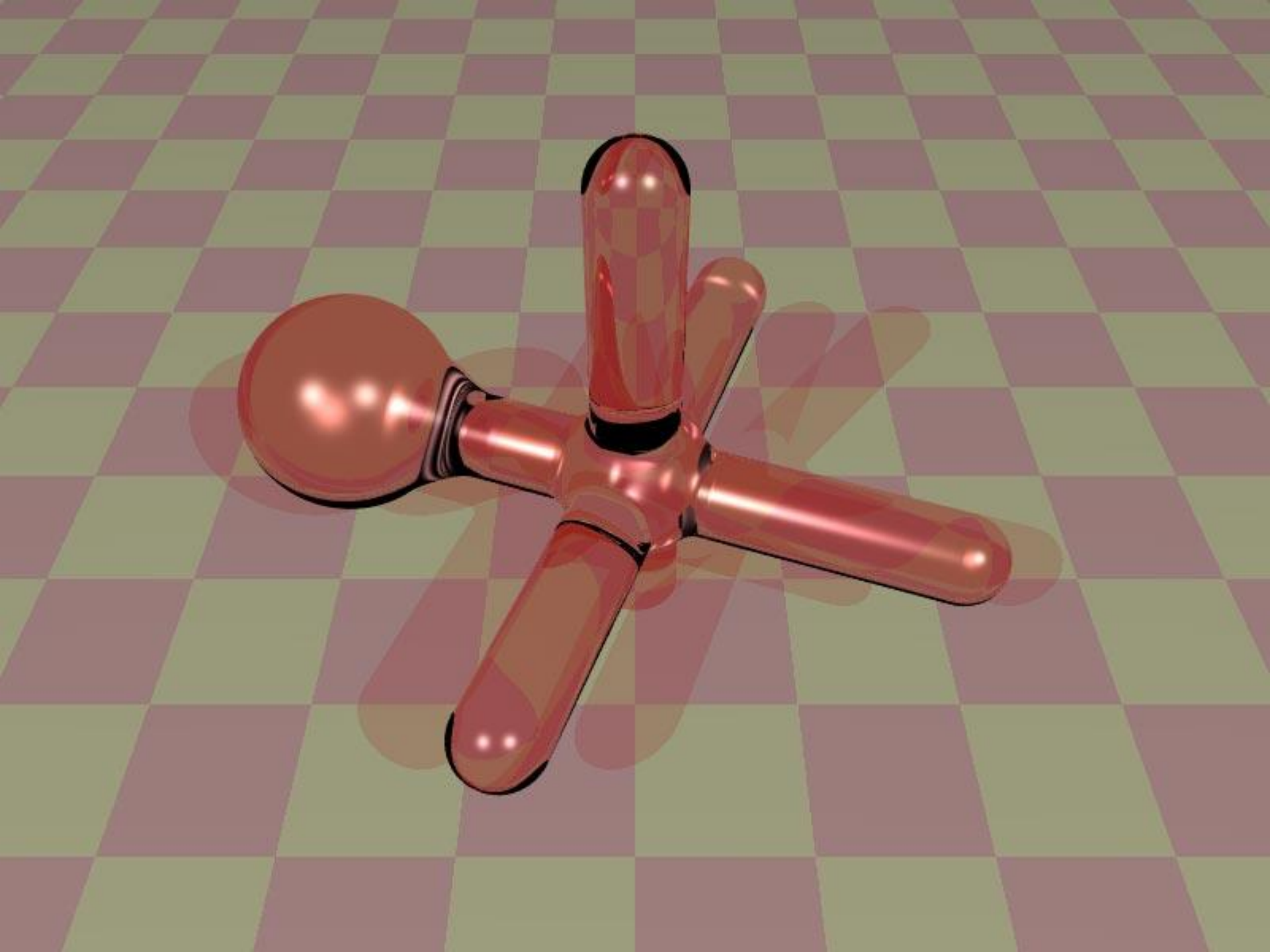










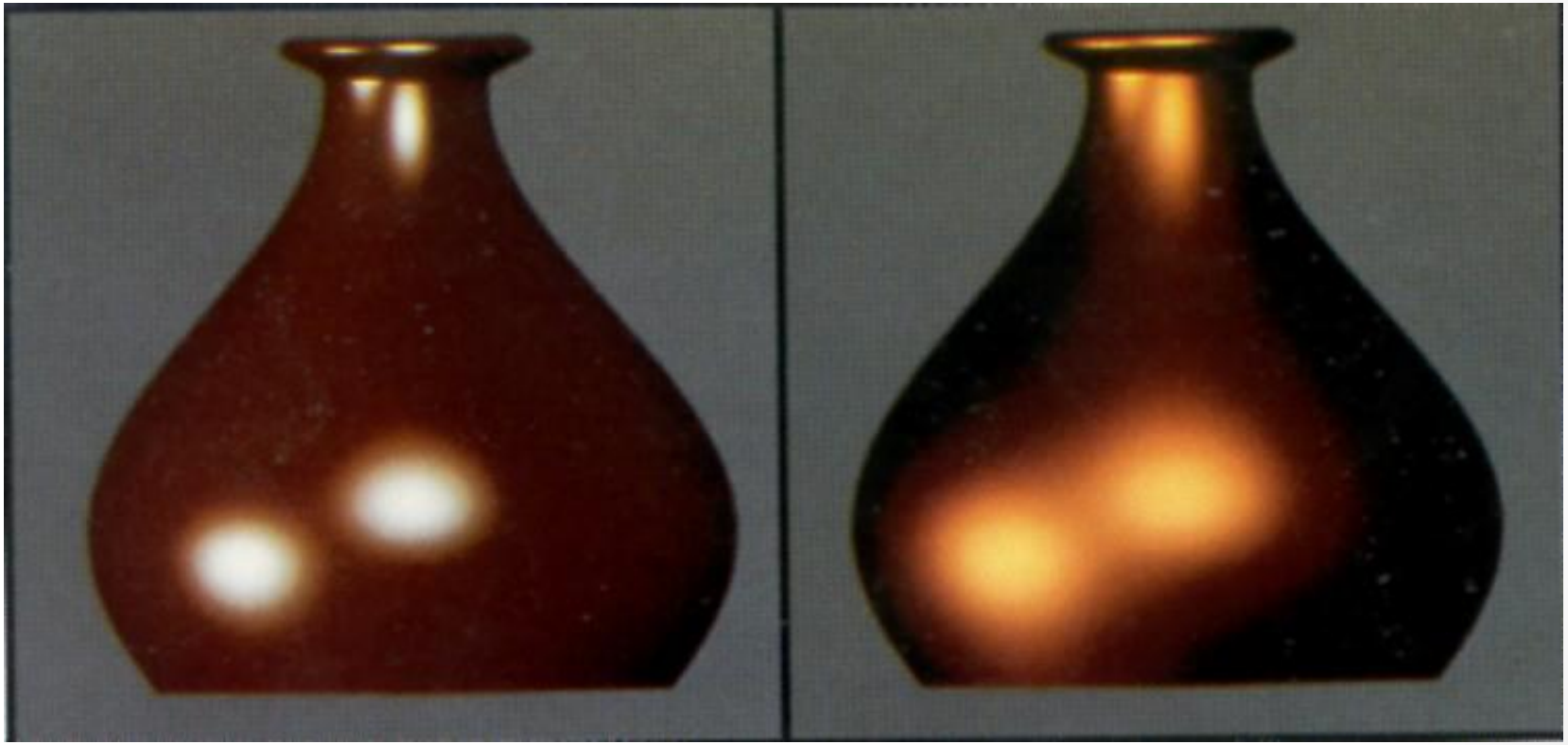


Types of illumination

- **Ambient** – "light soup" that affects every point equally
- **Diffuse** – shading that depends on the angle of the surface to the light source
- **Specular** – 'highlights.' Falls off sharply away from the reflection direction

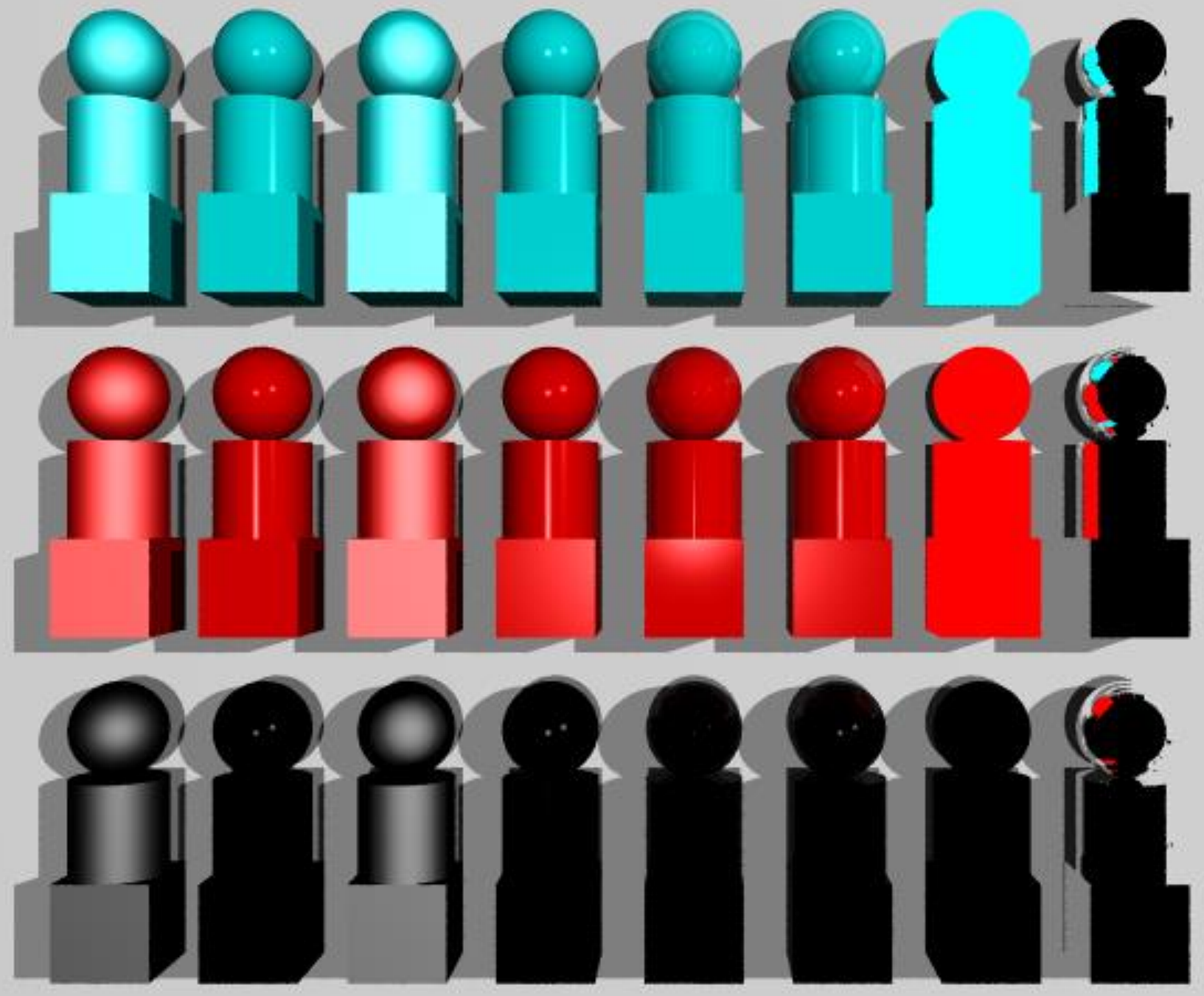
- Example: lighting applet

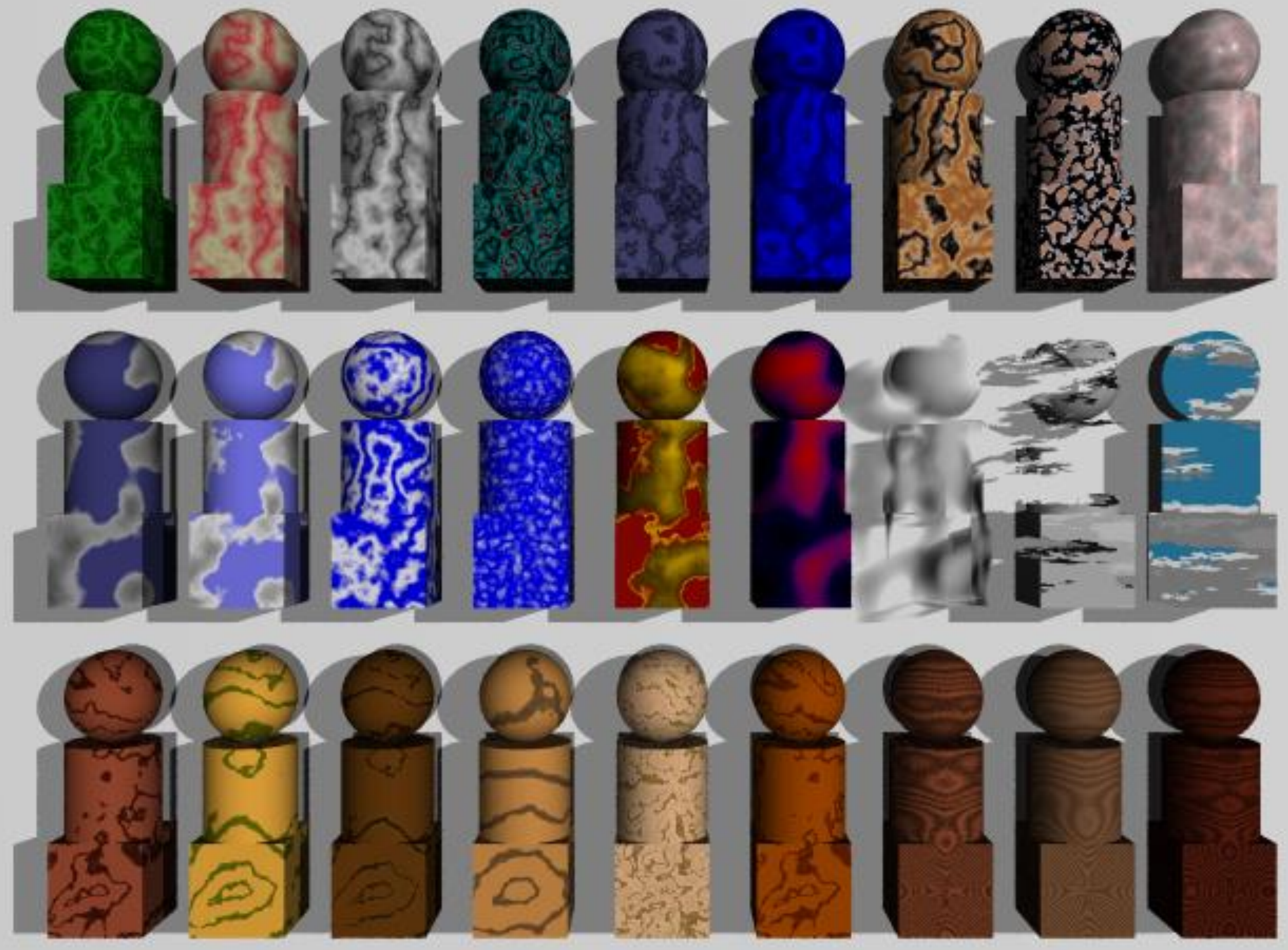
What are these made of?

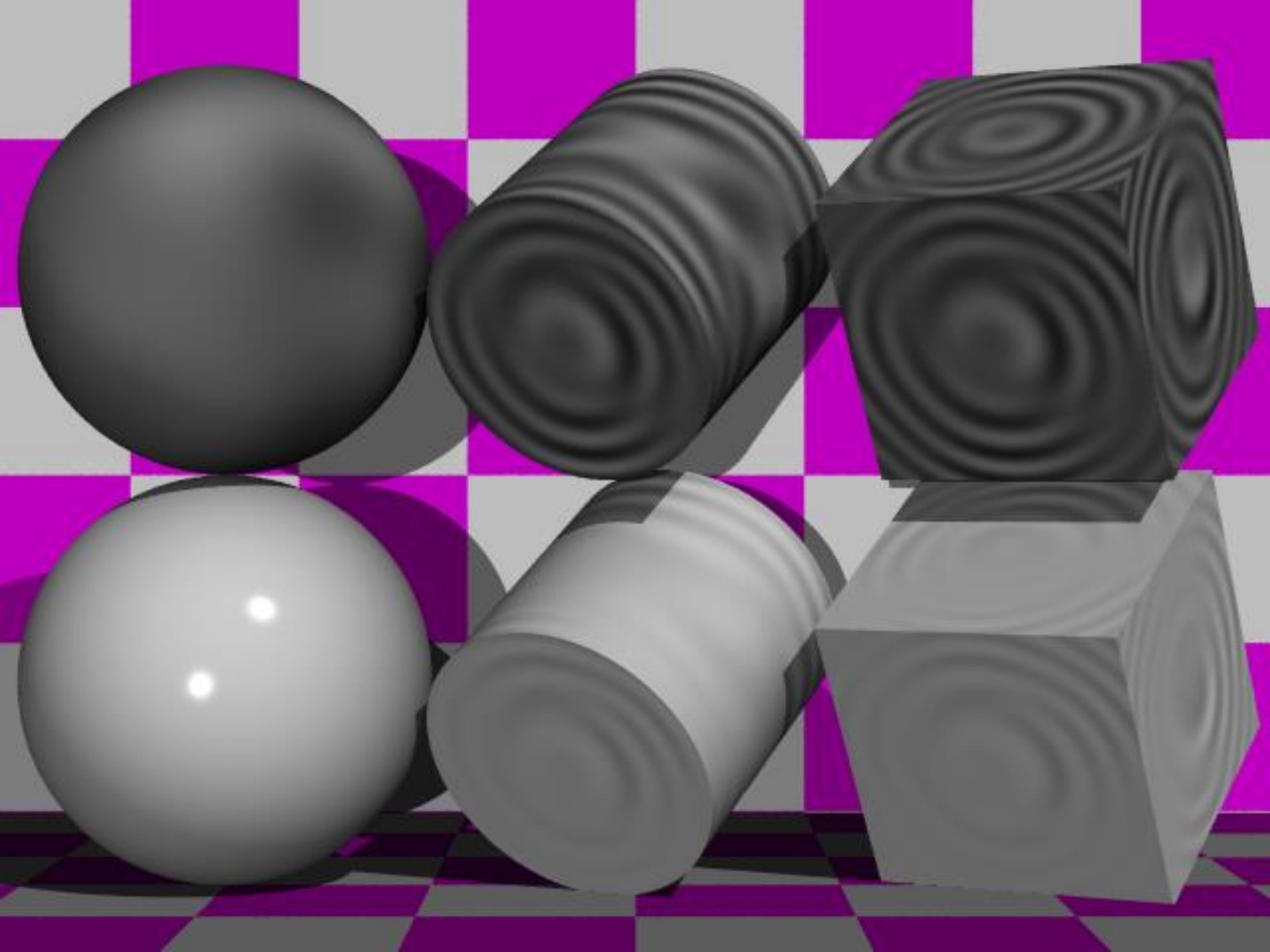


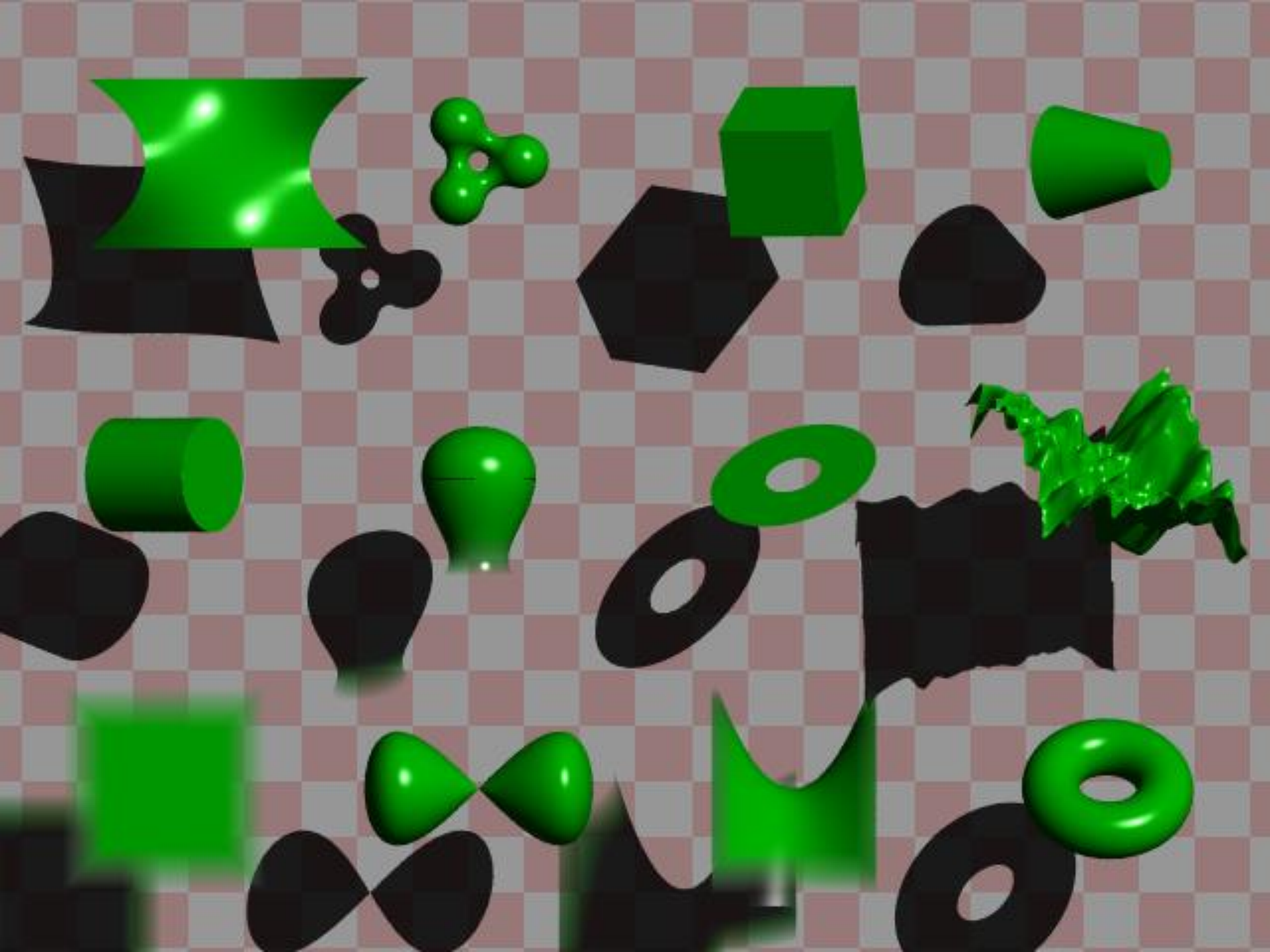
Material types

- Dielectrics (non-conductors):
  - In body reflection, light penetrates the surface and is affected by material pigment
  - Highlights are the color of the light source
  - Examples: paint, plastic, wood, …
- Conductors (metals)
  - No light penetrates the surface
  - Highlight and "body" reflection are affected equally by the material
  - Same color for diffuse and specular reflection









Constructive Solid Geometry

Plate III.2 Ray tracing CSG (Section 15.10.3). Bowl with stars and moons. (a) Objects used in CSG operations. (b) Cross-section of (a). (c) Bowl formed by CSG operations. Red sphere is truncated by intersection with blue cylinder, and result is hollowed by subtracting internal grey sphere visible in (b). Extruded moons (formed by subtracting green cylinders from orange cylinders) and extruded white stars cut holes in bowl when subtracted from earlier result. Reflection mapping (Section 16.6) and Cook–Torrance illumination model (Section 16.7) are used to give bowl a metallic appearance. (Courtesy of David Kurlander, Columbia University.)

How to ray-trace…

- Transparency?
- Refraction?
- Reflection?
- Fog?
- Anti-aliasing?

Drawbacks of ray tracing?

- Time: many rays are needed per pixel…
  - Up to 25 rays through each pixel
  - Each ray may bounce and split many times
  - Each ray tested for intersection with many objects
  - E.g. 1M pixels * 25 rays per pixel * 40 rays per ray tree * 1000 objects = 1 trillion object intersection tests…
  - Too slow for real time?
- Hard lighting
  - No soft shadows, inter-object diffusion, etc

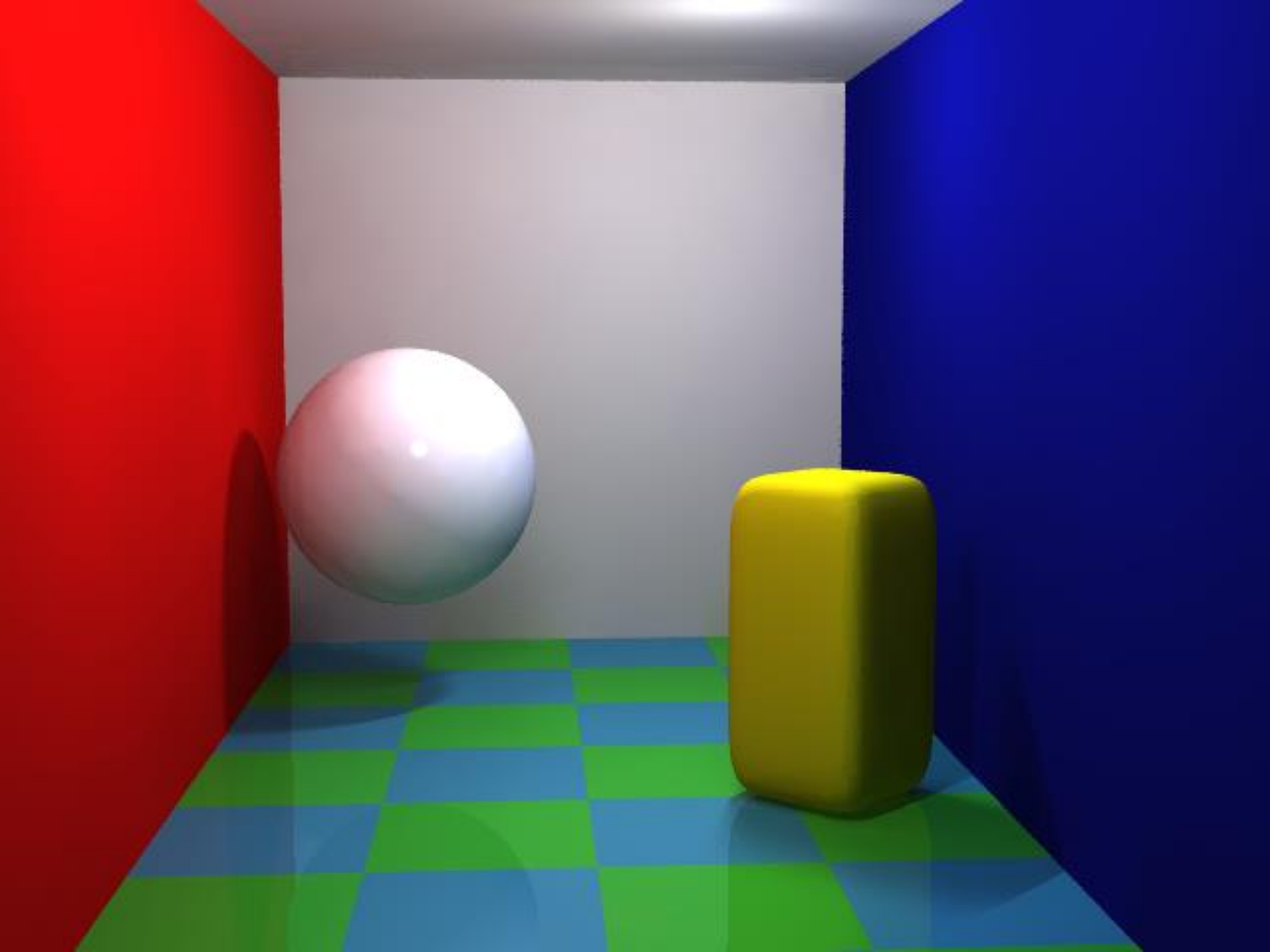
POV-Ray

- An excellent, free ray tracer: POV-Ray
- We'll use for a brief intro to ray tracing
- Runs on PC, Unix, Mac, Beowolf clusters, …
- Installed on the computers in the Unix lab
- You may wish to install on your own computer

- First "lab": make a ray-traced image of four different types of primitives, one each plastic, glass, metal, and mirrored, over checked floor

Beyond Ray Tracing

- Problems with ray tracing:
  - hard shadows
  - no color bleeding
  - slow

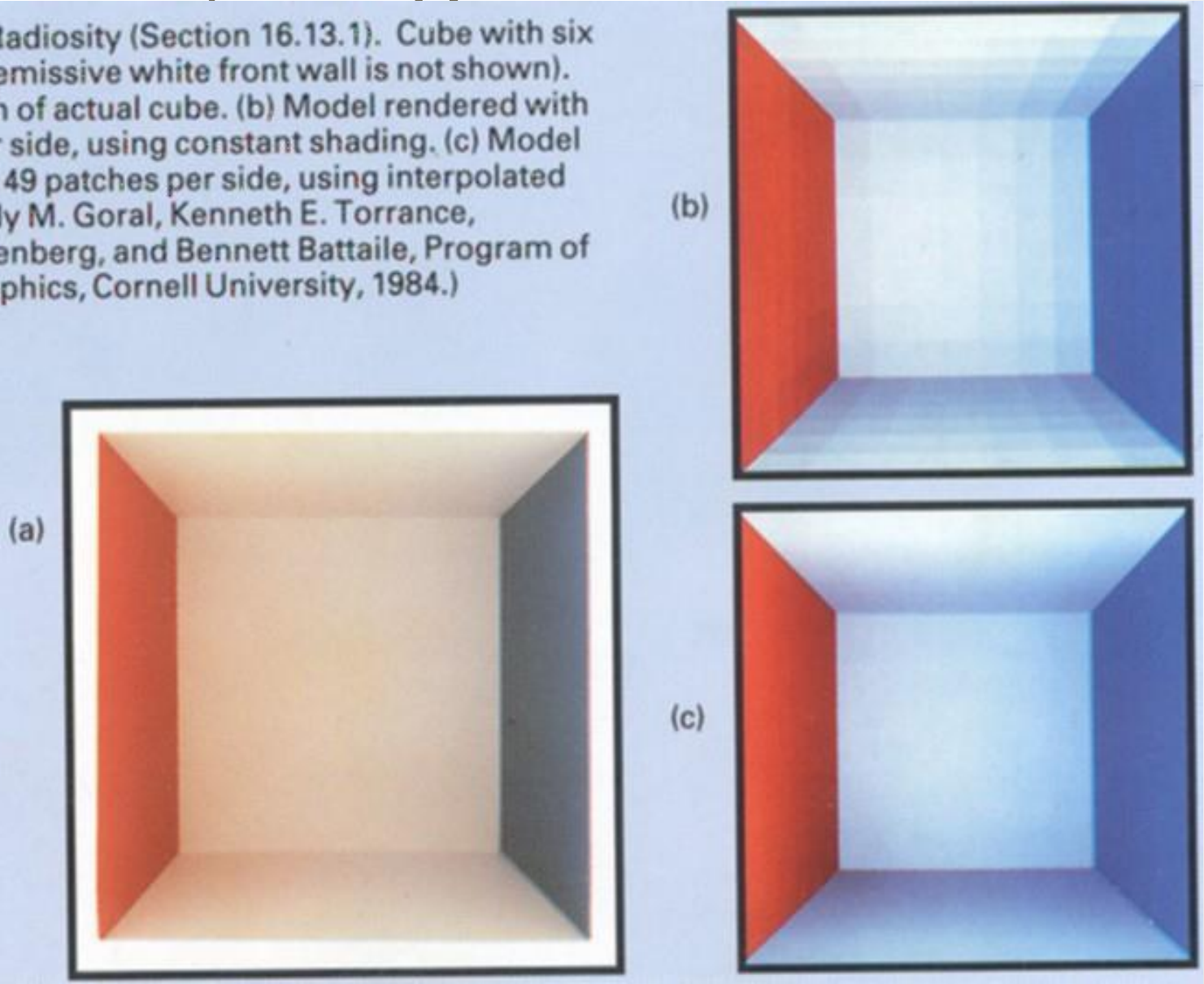


Radiosity

- Treat each patch as reflector and emitter of light
- Each patch affects every other patch depending on distance, orientation, occlusion etc.
- Let light "bounce around" for a few iterations to compute the amount of light reaching a patch

Radiosity image

Plate III.20 Radiosity (Section 16.13.1). Cube with six diffuse walls (emissive white front wall is not shown). (a) Photograph of actual cube. (b) Model rendered with 49 patches per side, using constant shading. (c) Model rendered with 49 patches per side, using interpolated shading. (Cindy M. Goral, Kenneth E. Torrance, Donald P. Greenberg, and Bennett Battaile, Program of Computer Graphics, Cornell University, 1984.)



Radiosity - table

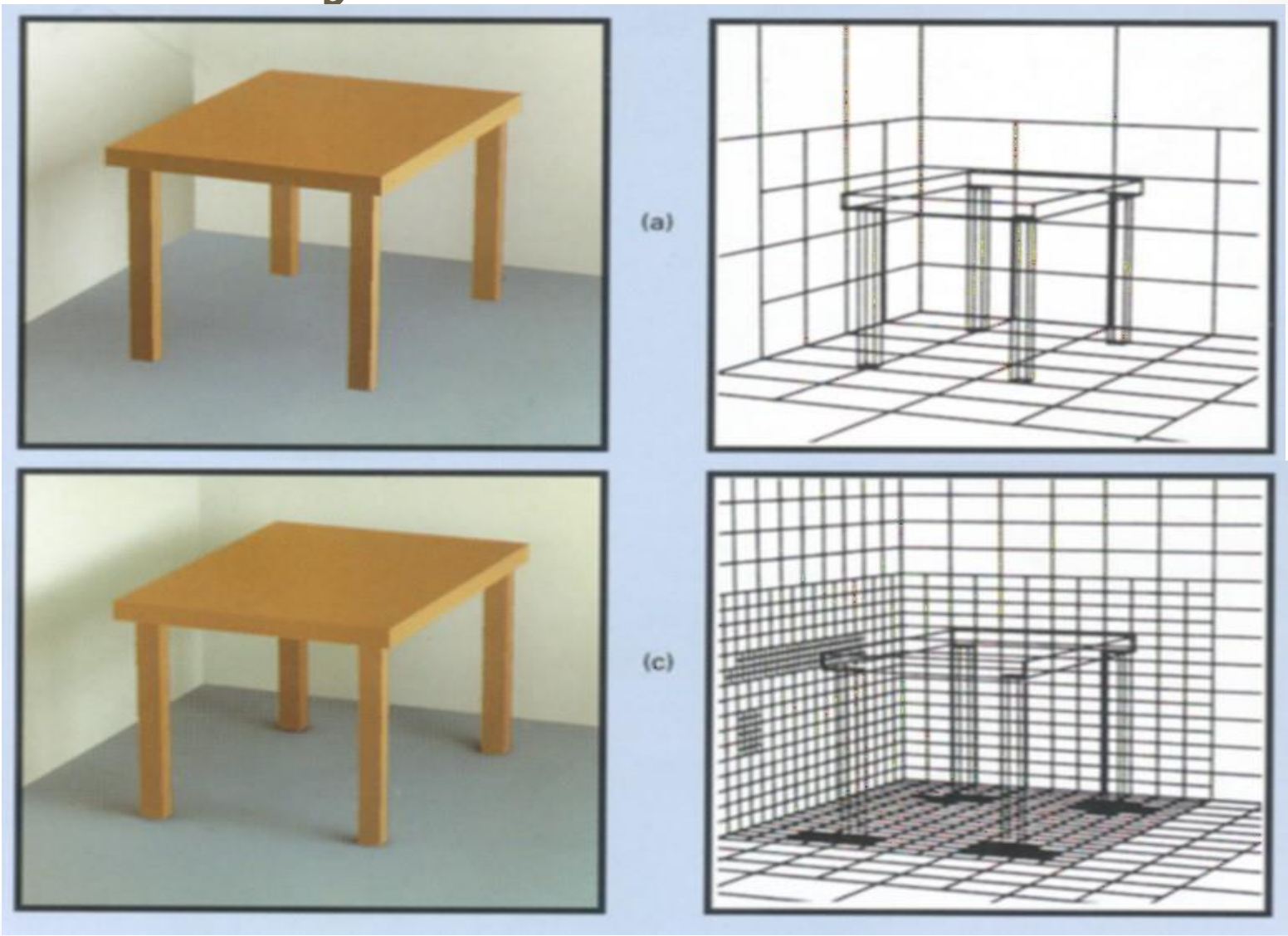


Image: Wikipedia

Radiosity example

The Radiosity Equation

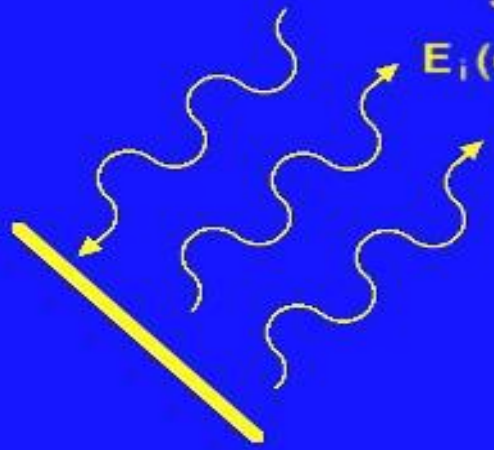
$$B_i = E_i + \rho_i \Sigma B_j F_{ij}$$

B_i = Radiosity of surface i
E_i = Emissivity of surface i
ρ_i = Reflectivity of surface i
B_j = Radiosity of surface j
F_{ij} = Form Factor of surface j relative to surface i

$\Sigma B_j F_{ij}$ (energy reaching this surface from other surfaces)

E_i (energy emitted by this surface)

$\rho_i \Sigma B_j F_{ij}$ (energy reflected by this surface)



Source: ACM

Radiosity summary

- Radiosity gives wonderful soft shading
- But even slower than ray tracing…
- Can't do reflection, refraction, specular highlights with radiosity
- Can *combine* ray tracing and radiosity for best of both worlds (and twice the time)

Interactive techniques

- Ray tracing and radiosity are too slow
- We'll concentrate on interactive techniques

- What kind of rendering can be done *quickly*?

Shutterbug - Orthographic

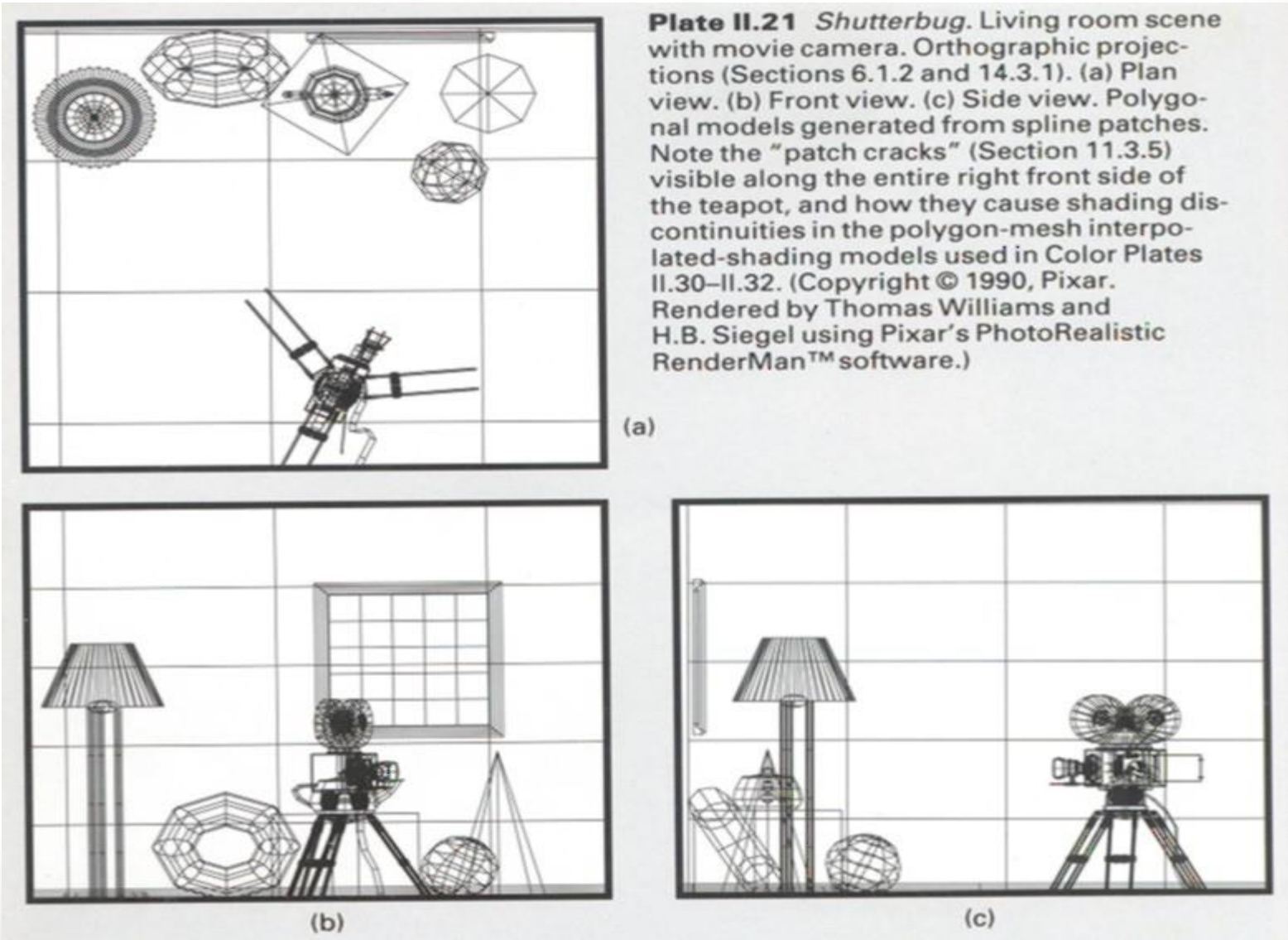
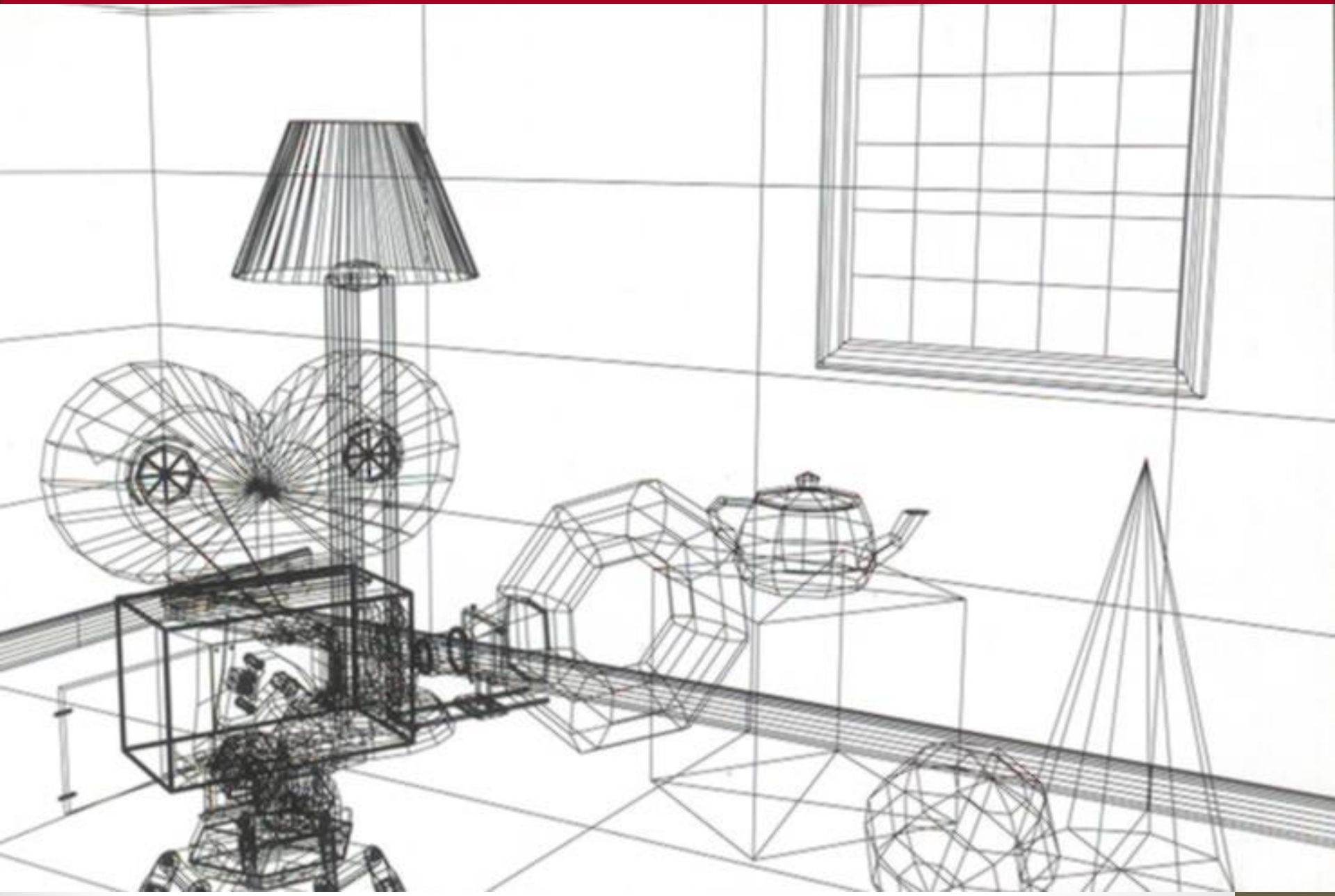


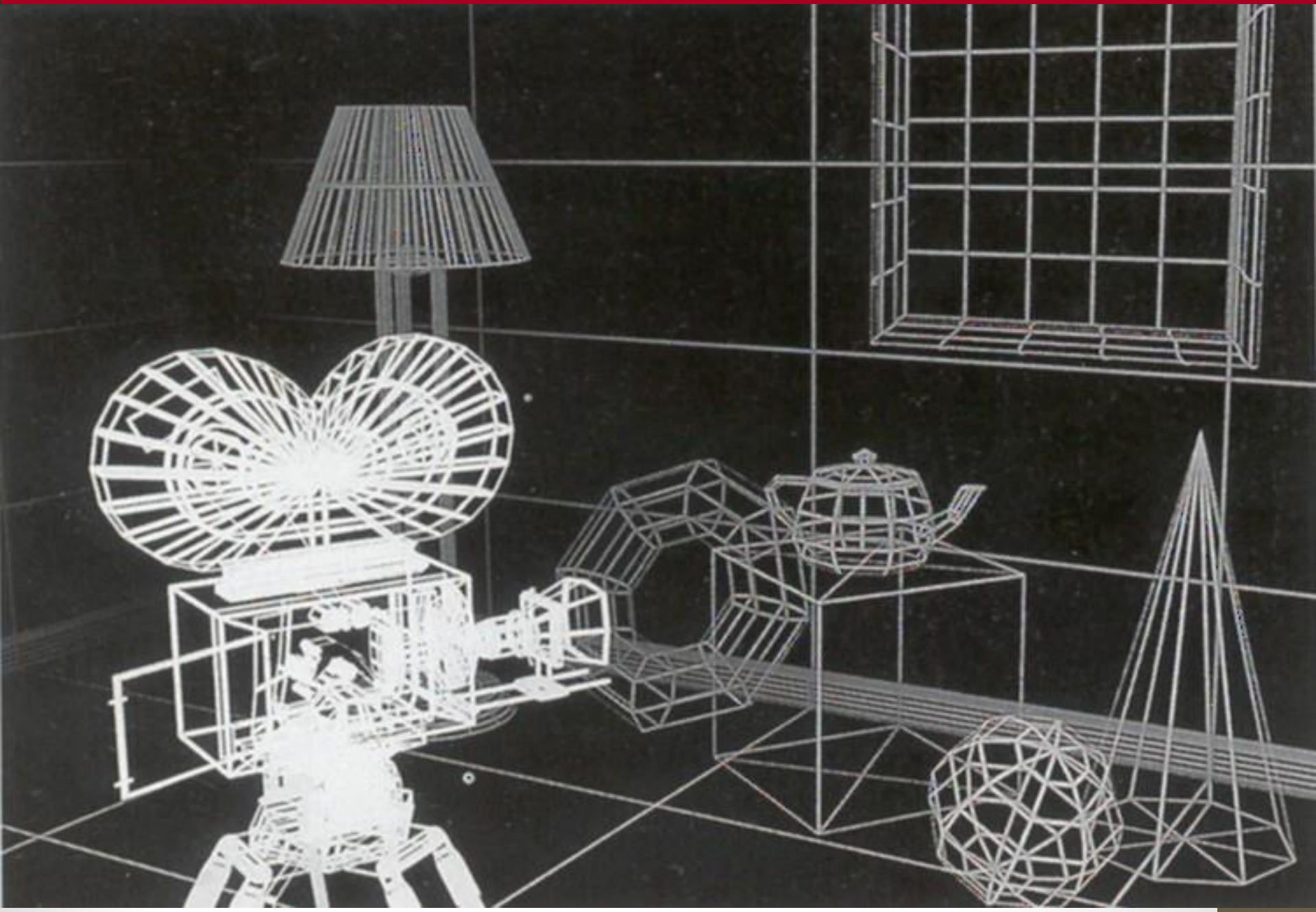
Plate II.21 *Shutterbug*. Living room scene with movie camera. Orthographic projections (Sections 6.1.2 and 14.3.1). (a) Plan view. (b) Front view. (c) Side view. Polygonal models generated from spline patches. Note the "patch cracks" (Section 11.3.5) visible along the entire right front side of the teapot, and how they cause shading discontinuities in the polygon-mesh interpolated-shading models used in Color Plates II.30–II.32. (Copyright © 1990, Pixar. Rendered by Thomas Williams and H.B. Siegel using Pixar's PhotoRealistic RenderMan™ software.)

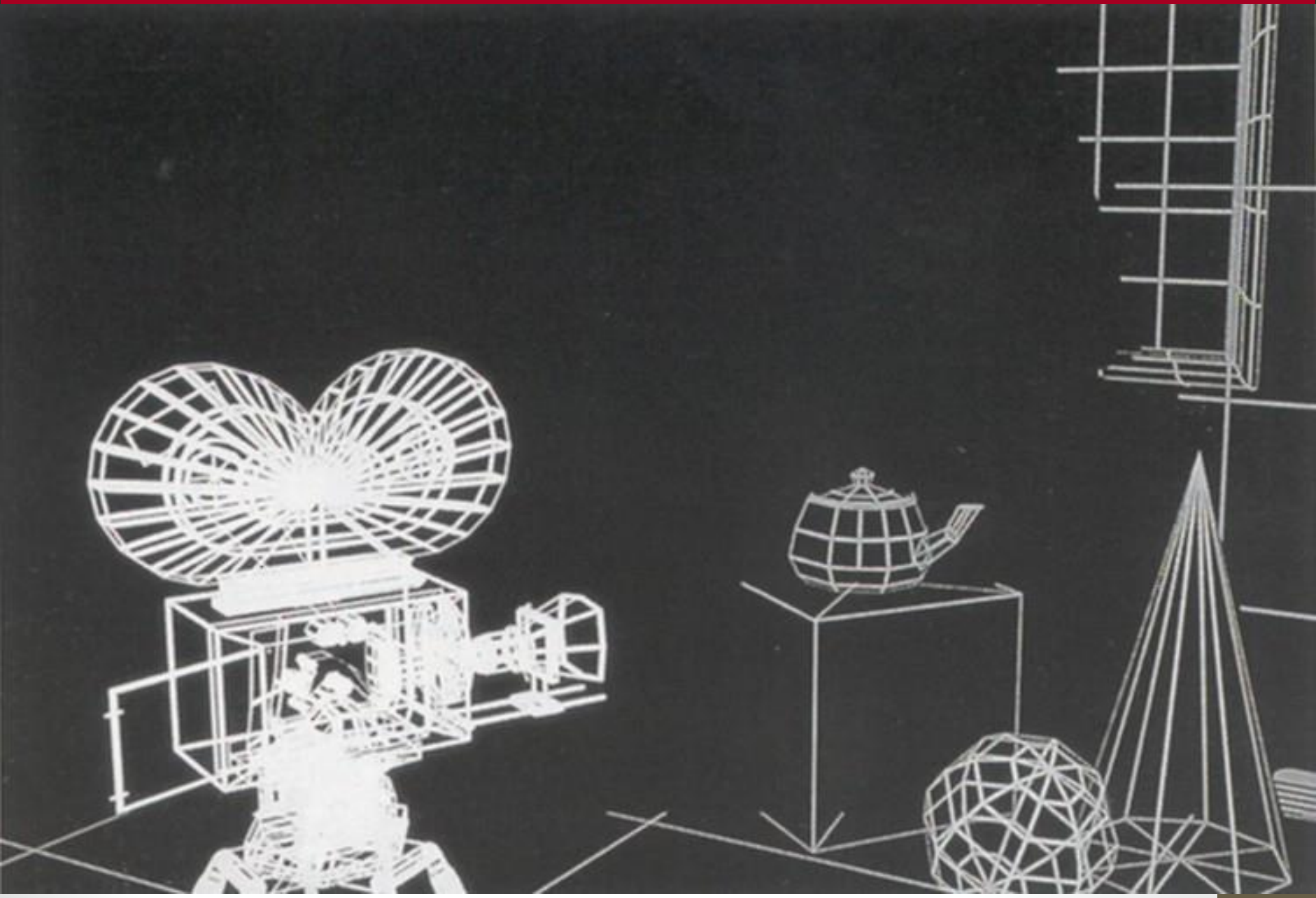
(a)

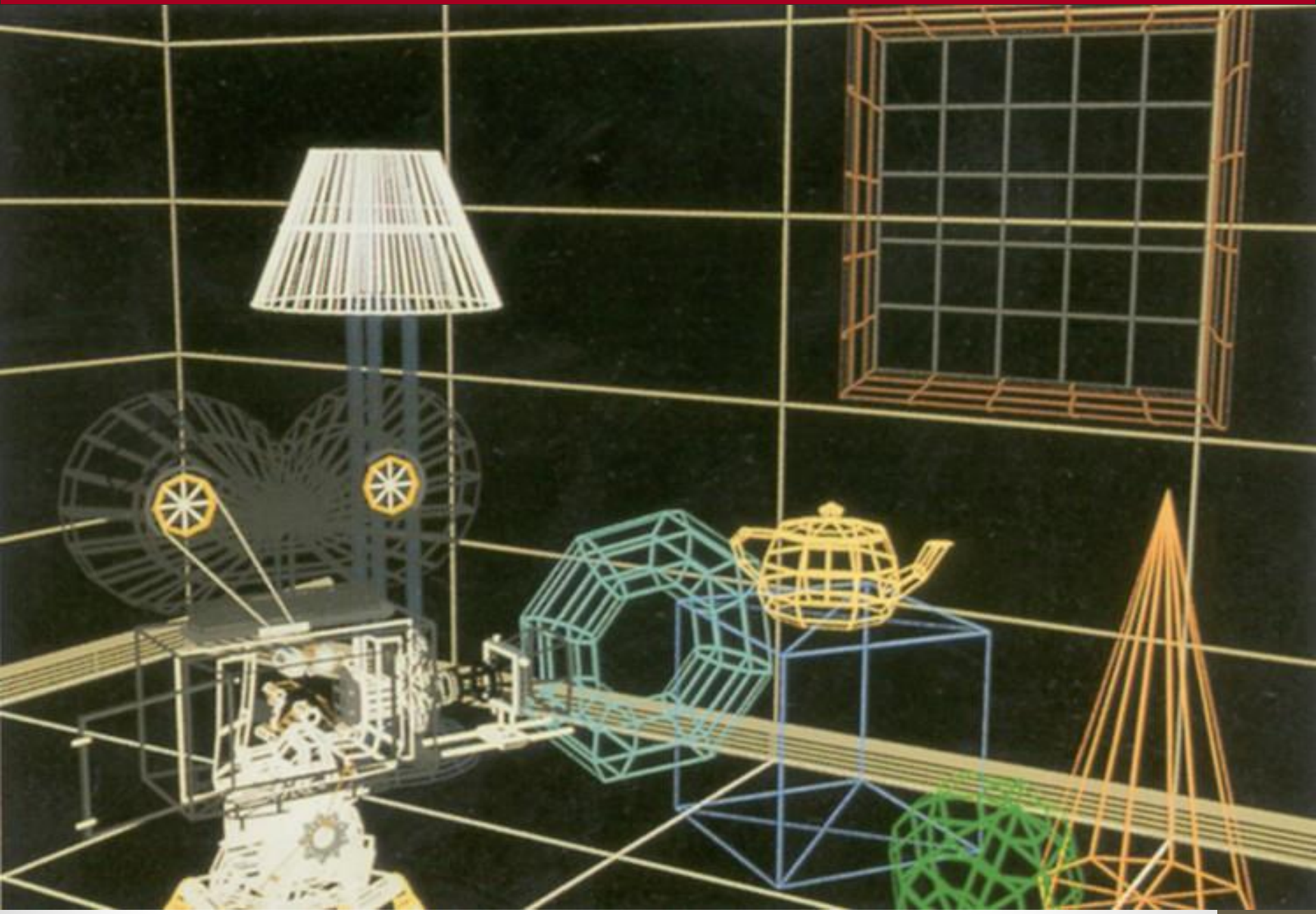
(b)

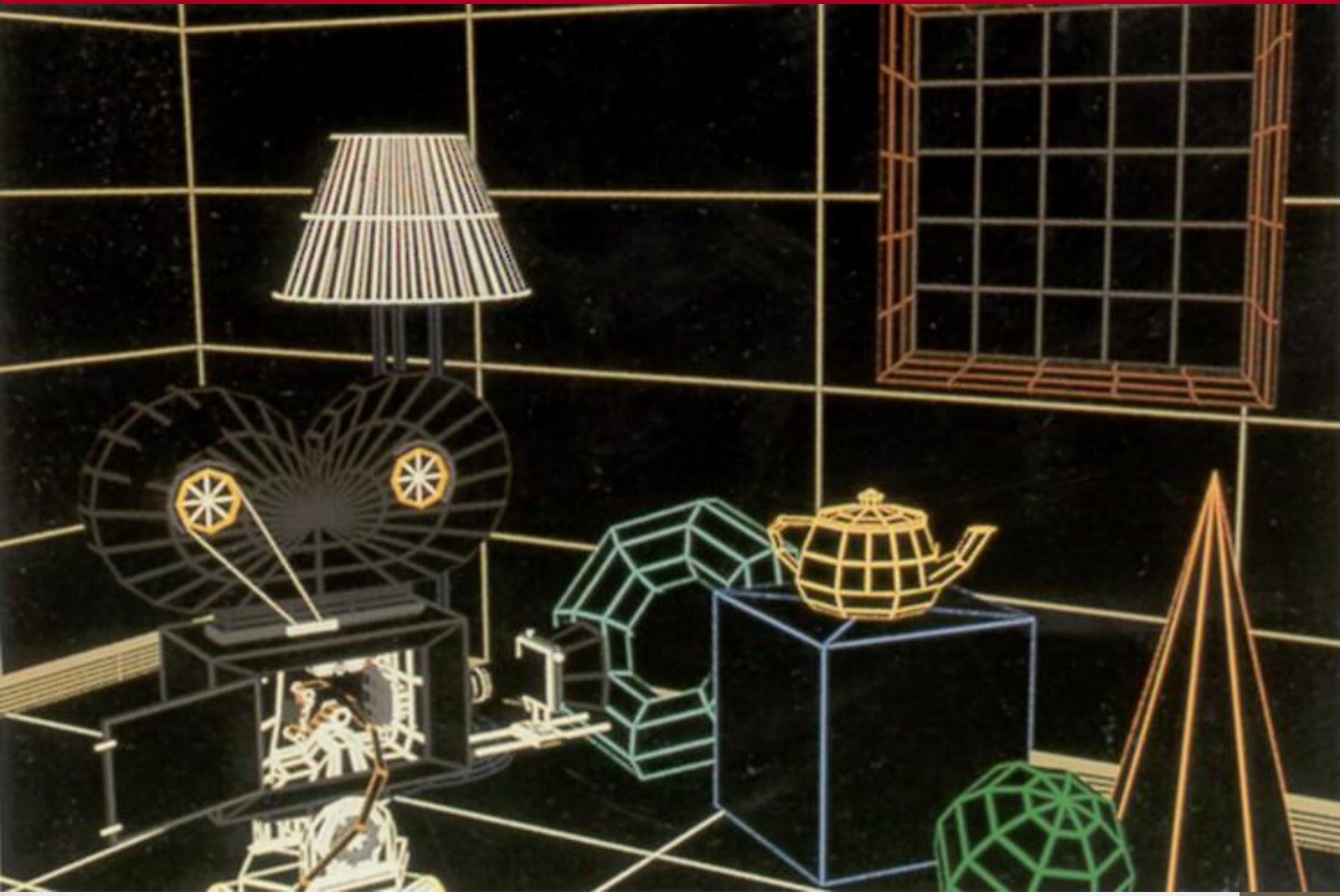
(c)

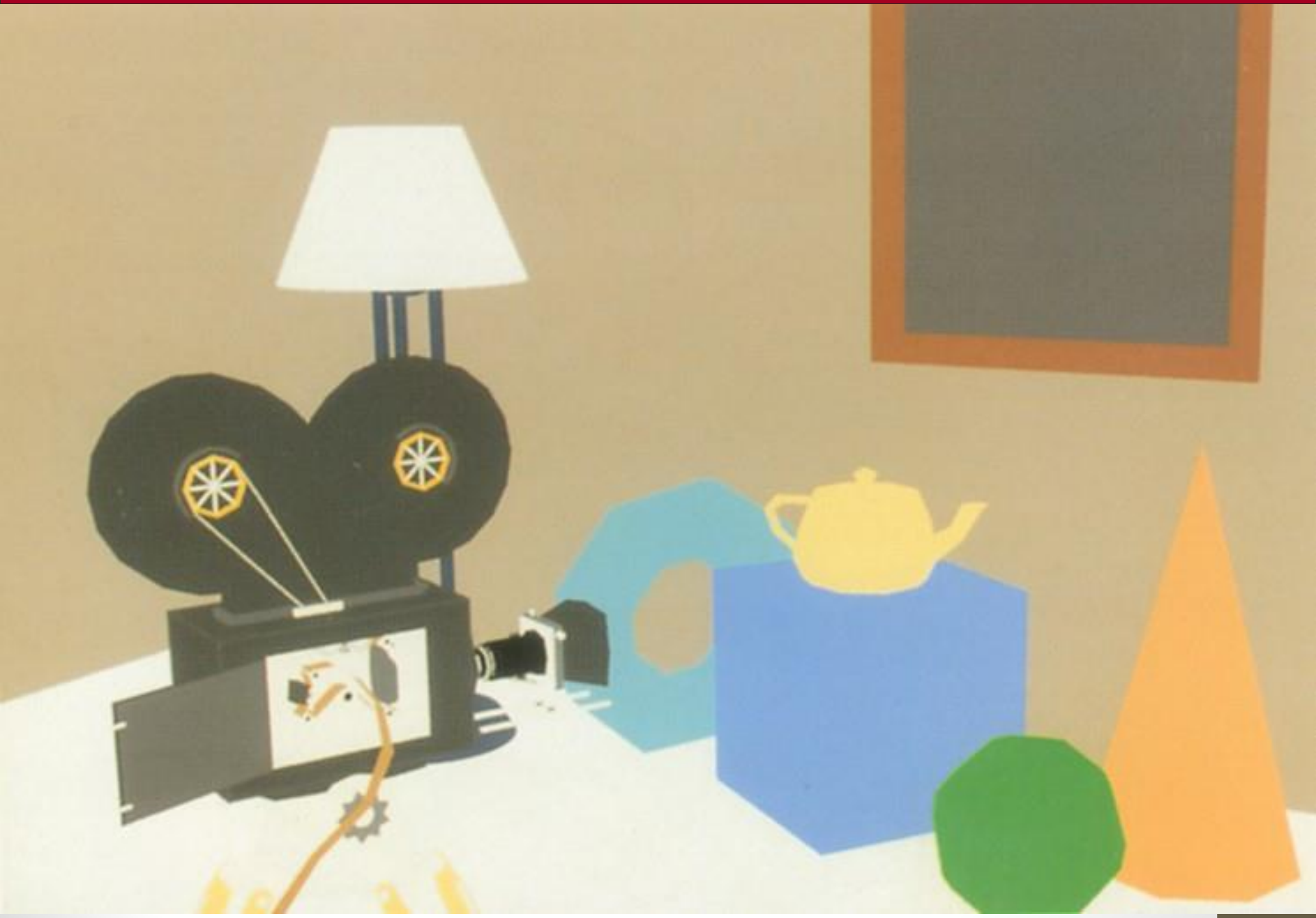


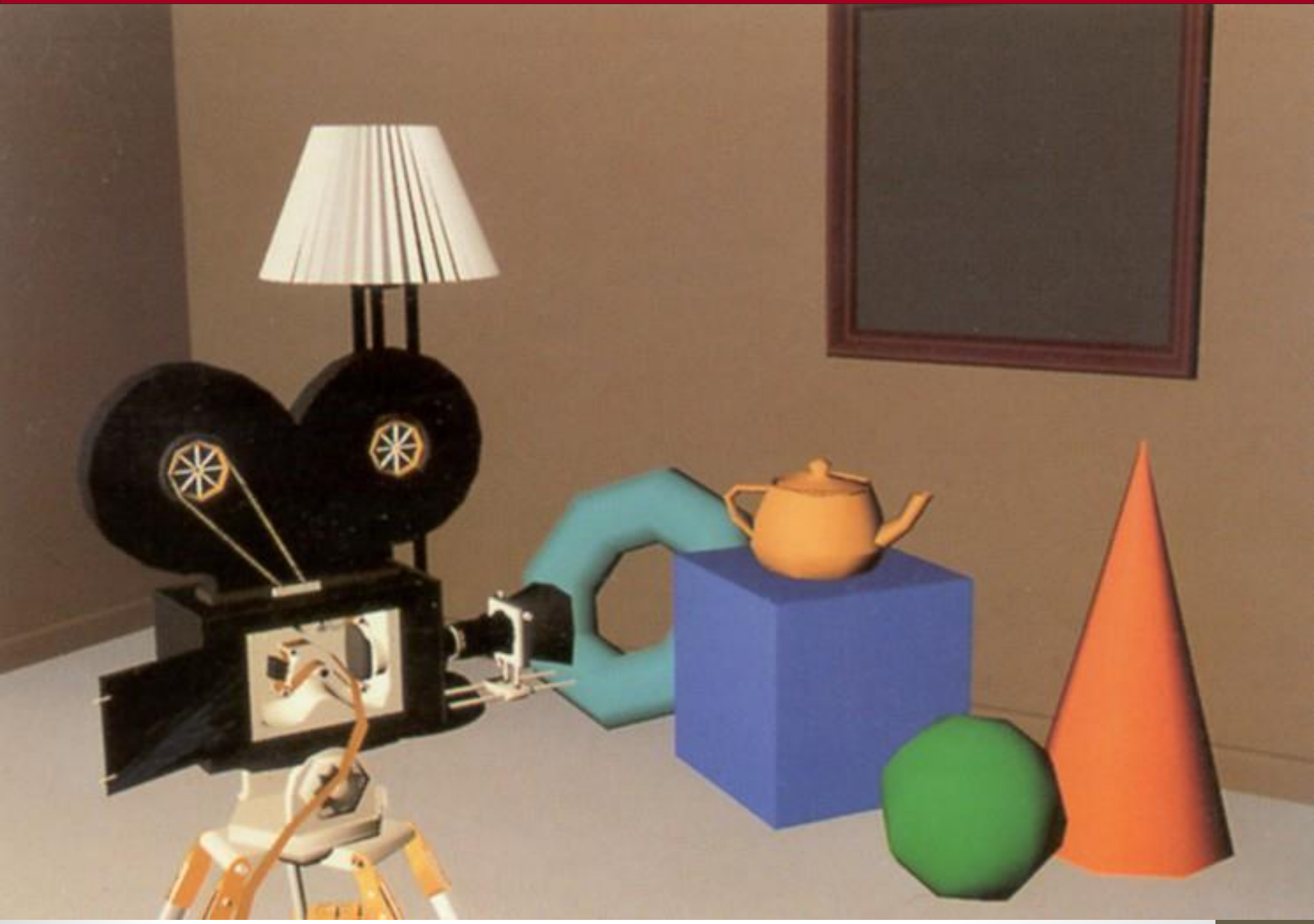


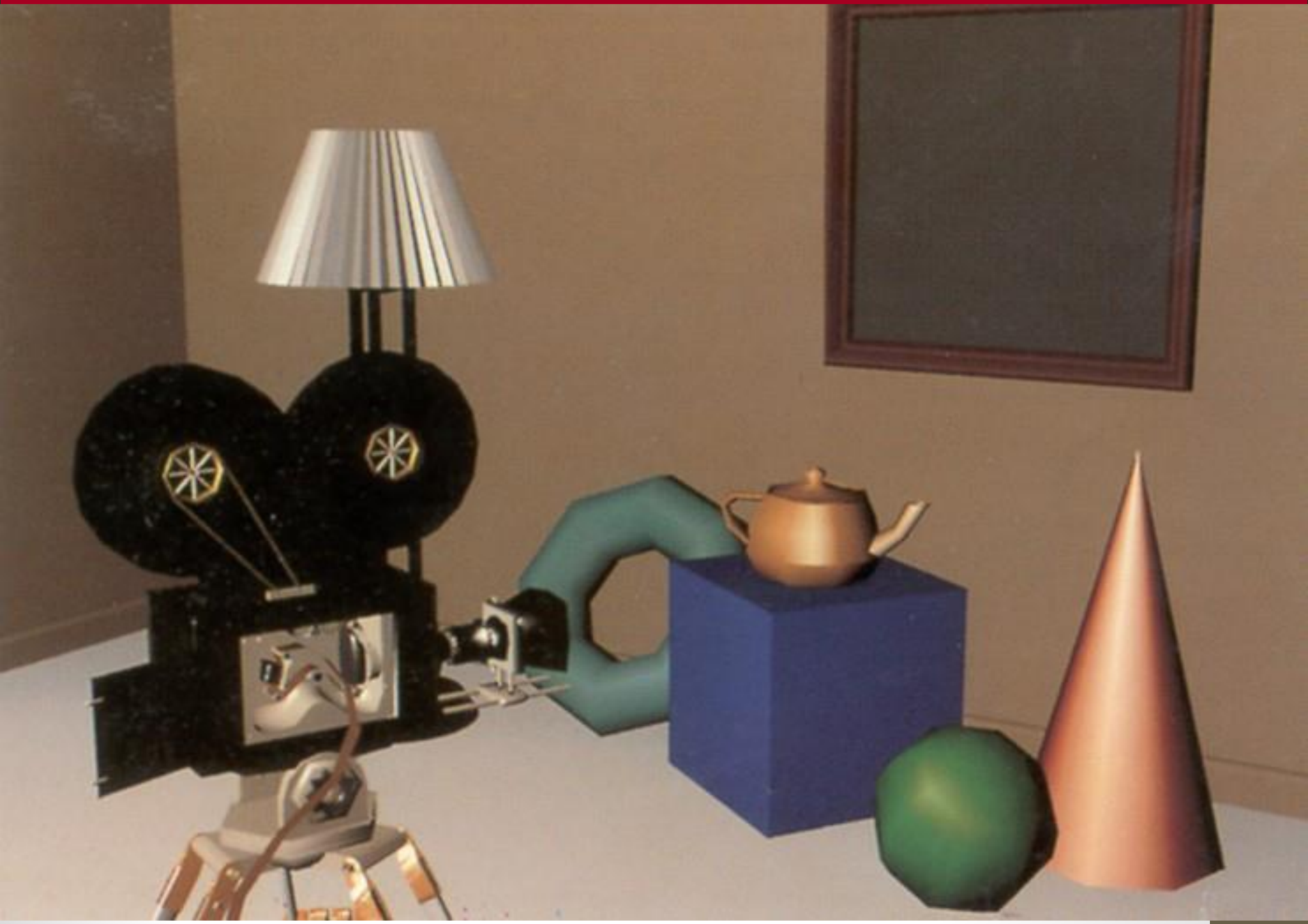


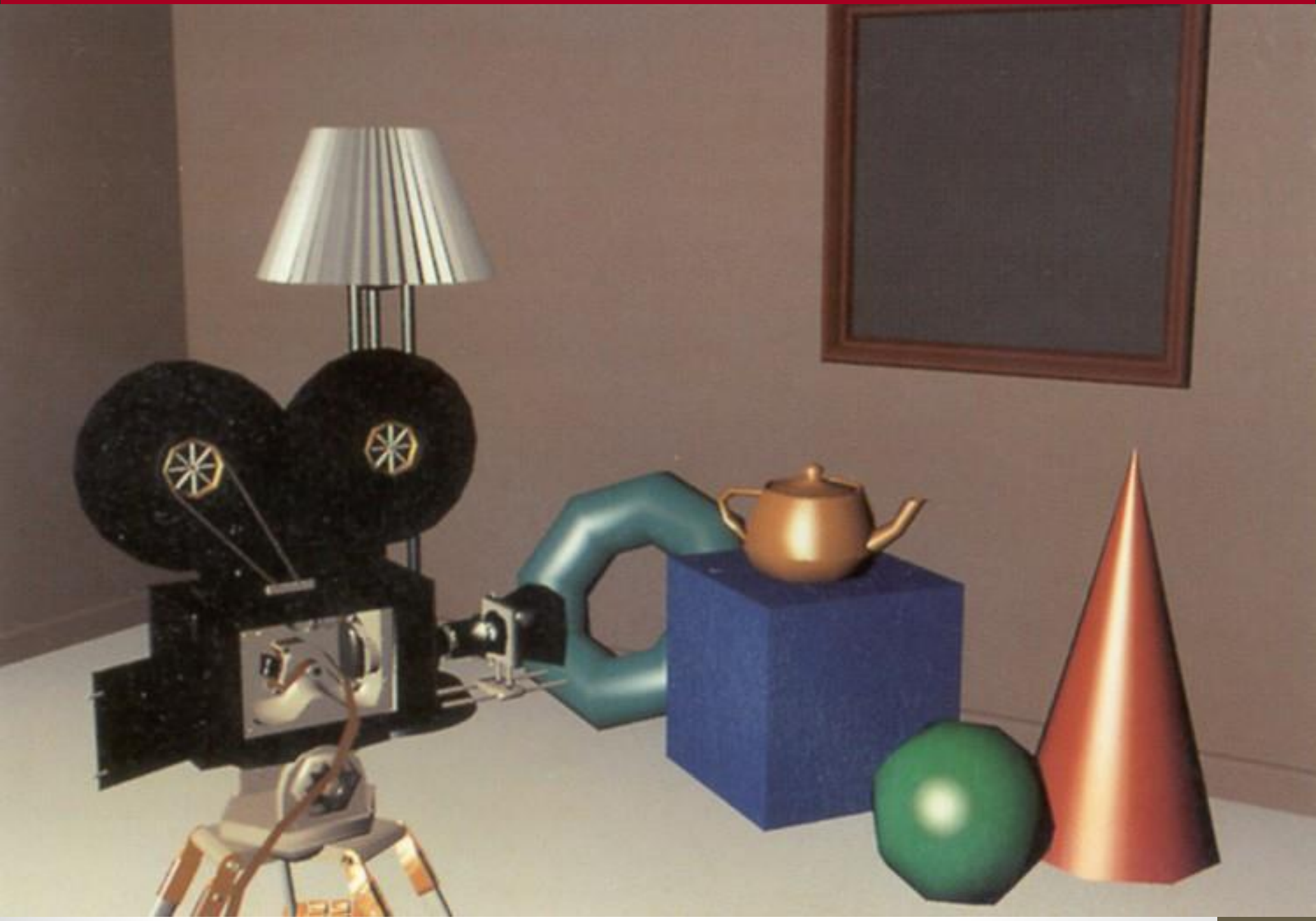


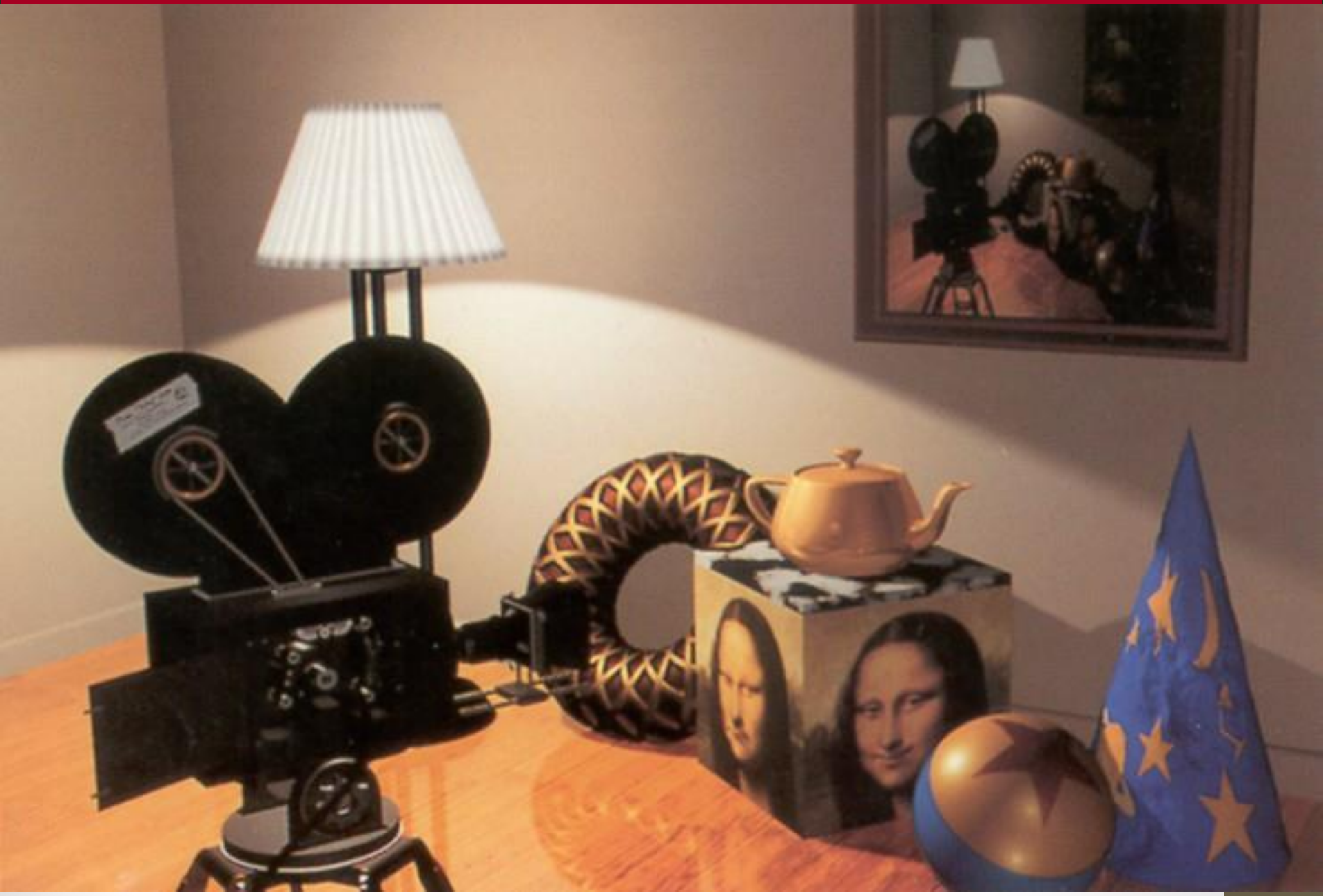




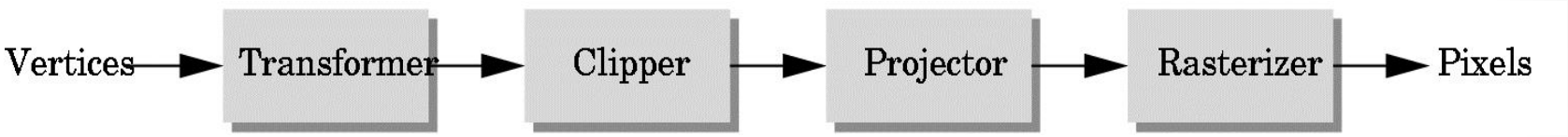








Rendering pipeline



- Transformations
- Clipping
- Projection
- Rasterization

- (what is done where?)