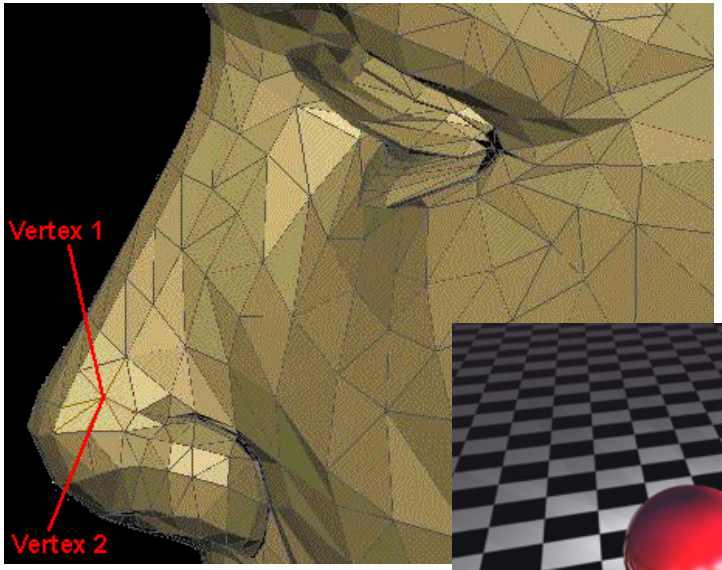
COMP250: Computer Graphics

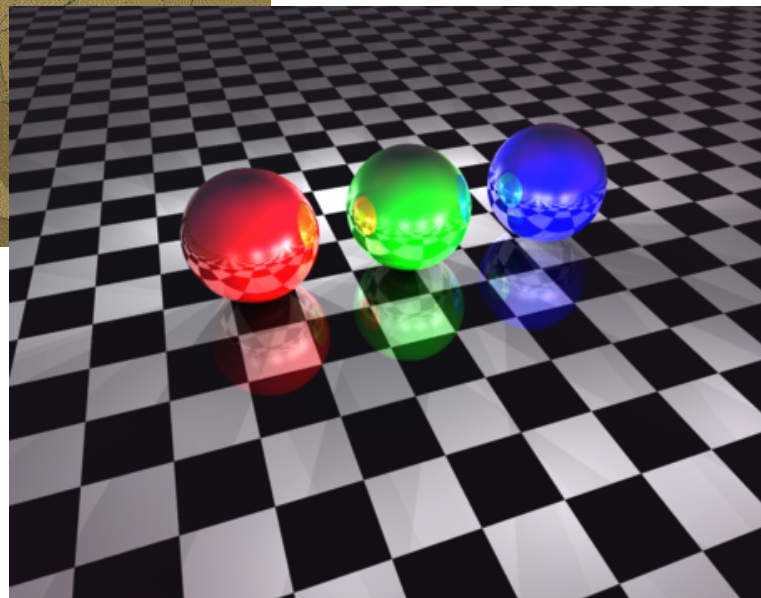
Jérôme Waldispühl
School of Computer Science
McGill University

Slides assembled from M. Blanchette (McGill), T. Thorne (U. of Edinburgh)

Computer Graphics Rendering

- World is represented by a set of 3D objects, with colors, reflectivity, transparence, etc.
  - Primite objects: Polygons, spheres, cones
  - Complex objects: Mesh of triangles





- Goal: Produce a realistic 2D picture of the world

Slide by M. Blanchette (McGill)

Computer Graphics is about animation (films)

Slide by T. Thorne (U. of Edinburgh)

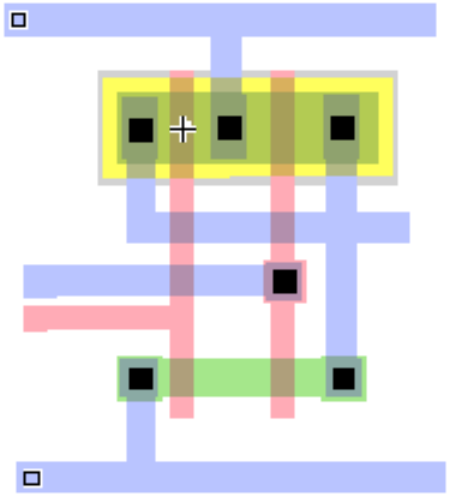
Games are very important in Computer Graphics

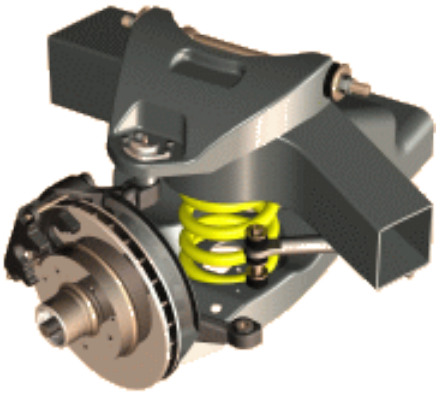
Slide by T. Thorne (U. of Edinburgh)

Medical Imaging is another driving force

Slide by T. Thorne (U. of Edinburgh)

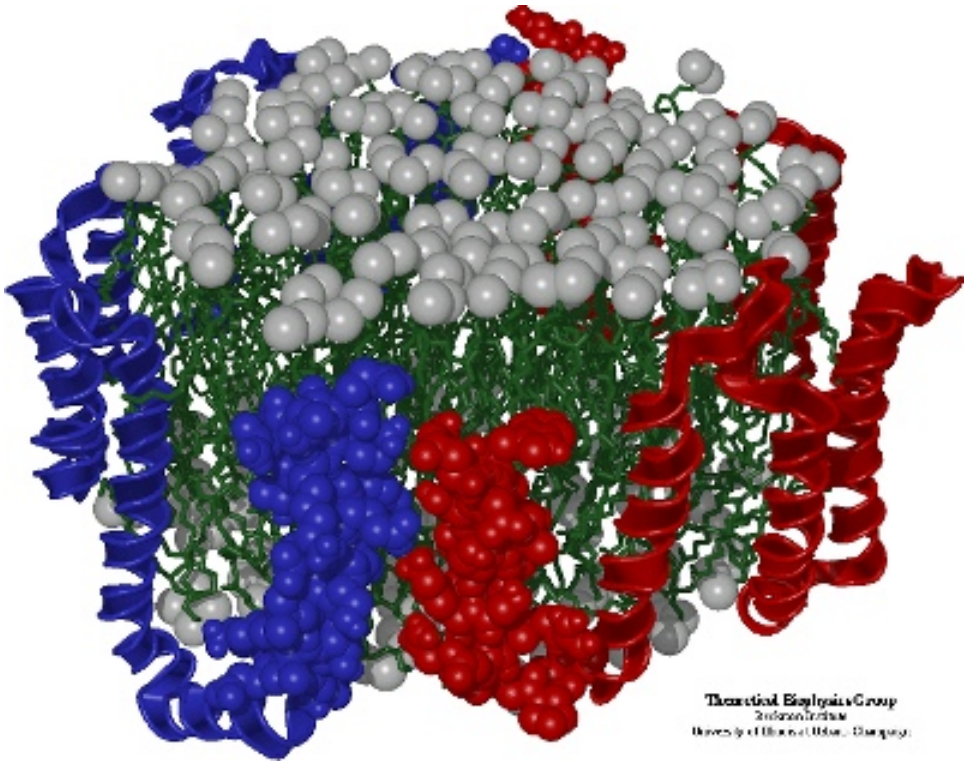
Computer Aided Design too

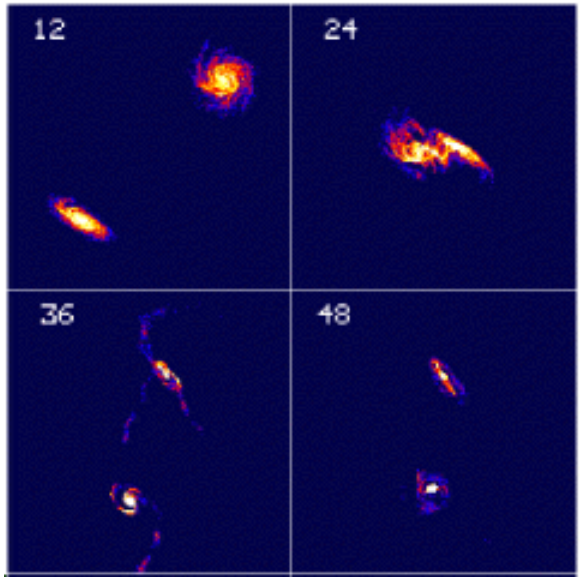




Slide by T. Thorne (U. of Edinburgh)

Scientific Visualisation



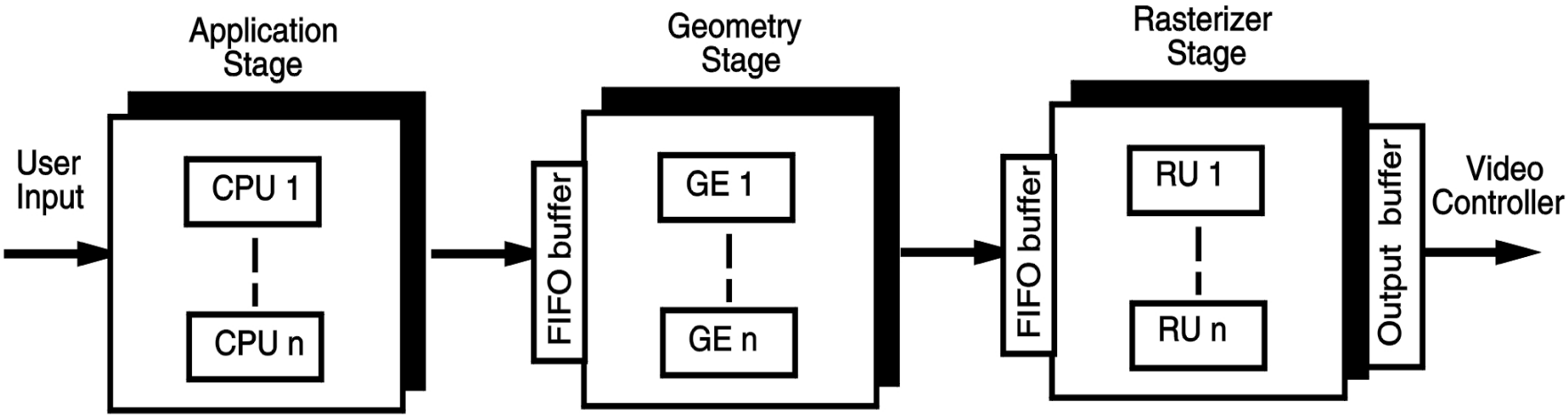


Slide by T. Thorne (U. of Edinburgh)

Graphics / Rendering Pipeline

There are three stages

- Application Stage
- Geometry Stage
- Rasterization Stage



Slide by T. Thorne (U. of Edinburgh)

Batch Computer Graphics

- Today, still use non-interactive *batch mode* for final production-quality video and film (special effects – FX). Rendering a single frame of Monsters University (a 24 fps movie) averaged 29 hours on a 24,000-core render farm!

Still from Monsters University

Render farm

Slide by A. van Dam (Brown U.)

Application stage

(for interactive computer graphics)

Entirely done in software by the CPU

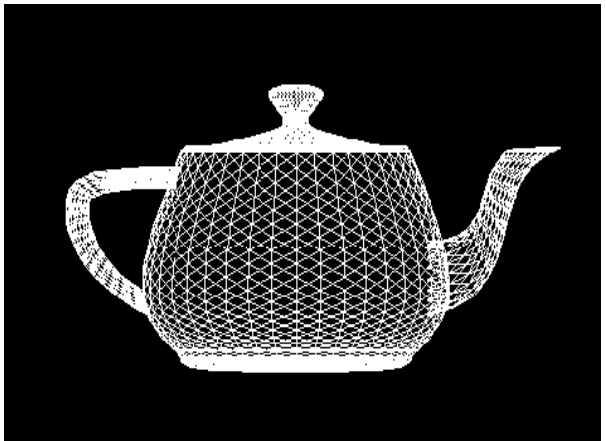
Read Data

- the world geometry database,
- User's input by mice, trackballs, trackers, or sensing gloves

In response to the user's input, the application stage change the view or scene

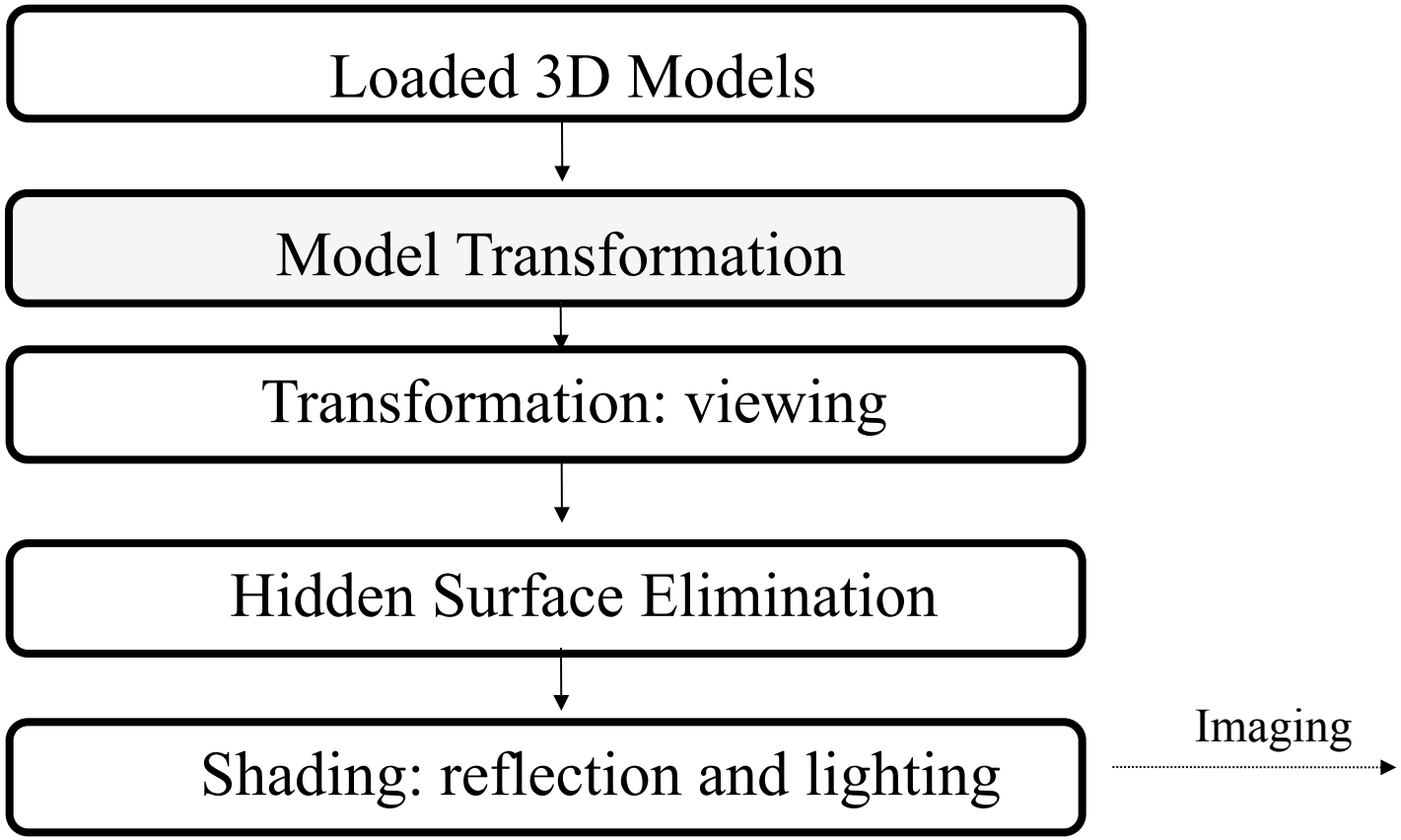
```
v 3.382035 2.446498 -0.064692
v 3.382035 2.446498 0.064692
v 3.392006 2.474995 -0.050004
v 3.392006 2.474995 0.050004
v 3.400000 2.446800 0.000000
v 3.406947 2.462176 -0.061668
v 3.406947 2.462176 0.061668
v 3.408000 2.475600 0.000000
v 3.408000 2.475600 0.000000
v 3.411237 2.476753 -0.054000
v 3.411237 2.476753 0.054000
v 3.416450 2.472371 -0.057996
v 3.416450 2.472371 0.057996
v 3.424875 2.462506 0.000000
v 3.428125 2.477344 0.000000
v 3.428125 2.477344 0.000000
v 3.434000 2.472900 0.000000

f 2909 2921 2939
f 2939 2931 2909
f 2869 2877 2921
f 2921 2909 2869
f 2819 2827 2877
f 2877 2869 2819
f 2737 2747 2827
f 2827 2819 2737
f 2669 2673 2747
f 2747 2737 2669
f 2567 2575 2673
f 2673 2669 2567
```



Slide by T. Thorne (U. of Edinburgh)

Geometry Pipeline

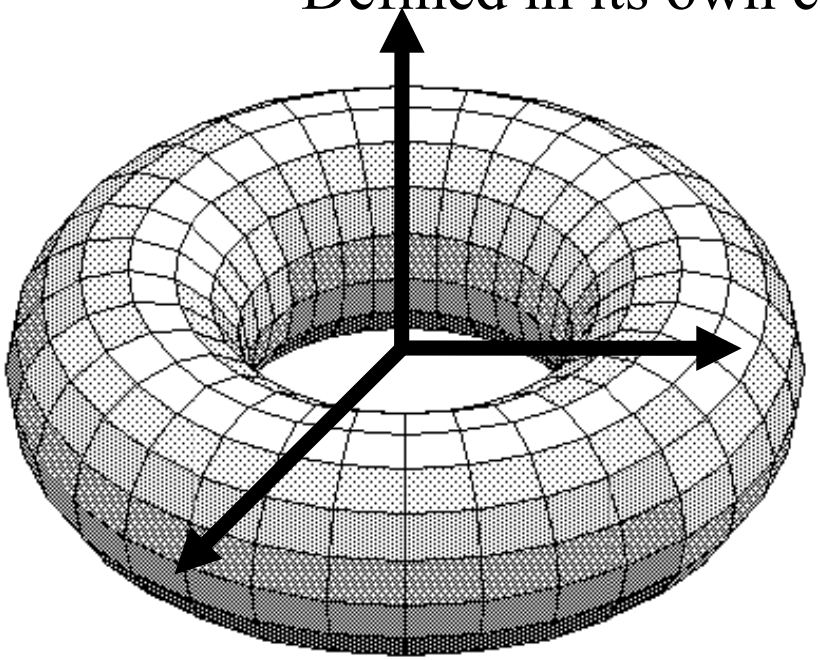


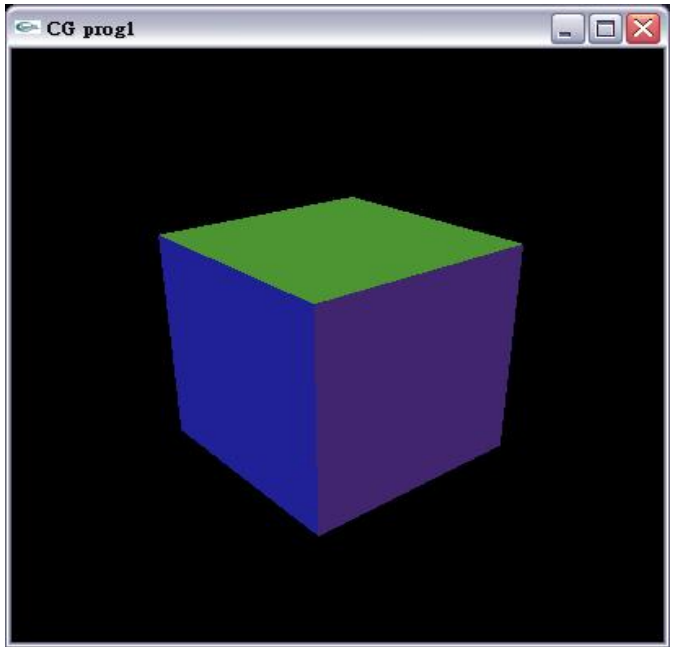
Slide by T. Thorne (U. of Edinburgh)

Preparing Shape Models

Designed by polygons, parametric curves/surfaces, implicit surfaces and etc.

Defined in its own coordinate system





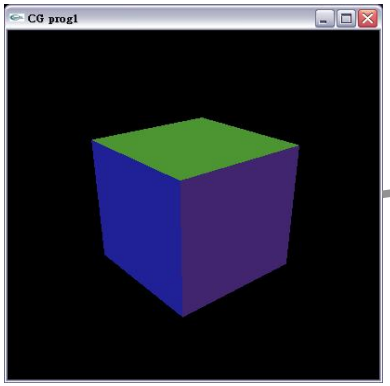
Slide by T. Thorne (U. of Edinburgh)

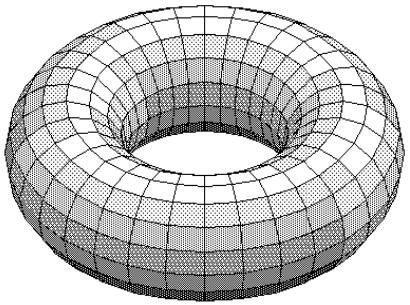
Model Transformation

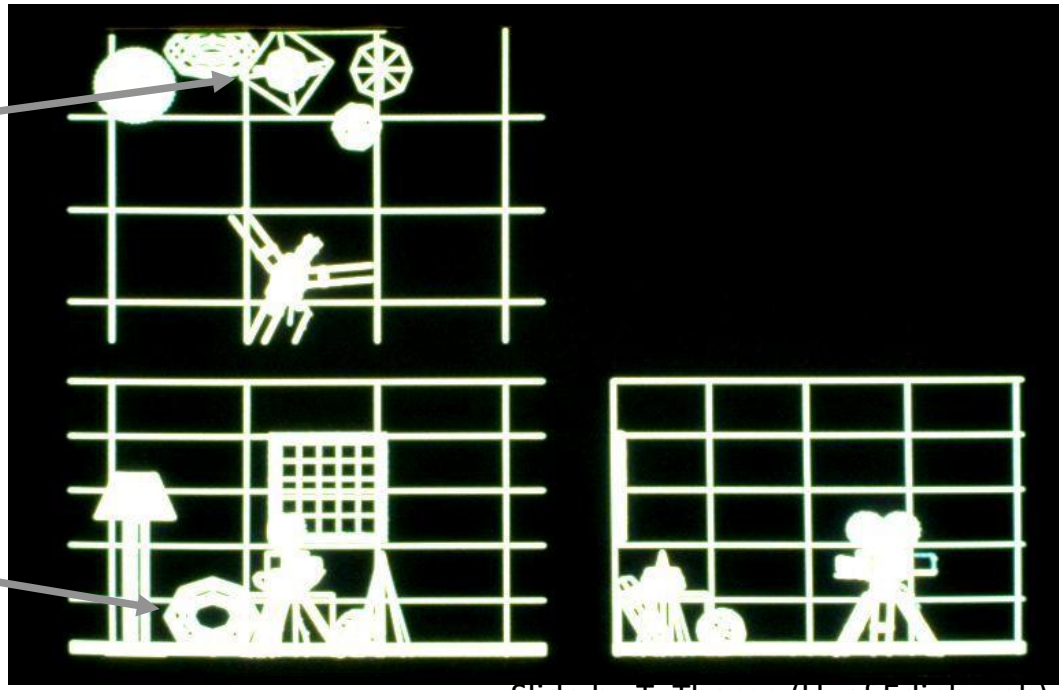
Objects put into the scene by applying translation, scaling and rotation

Linear transformation called homogeneous transformation is used

The location of all the vertices are updated by this transformation

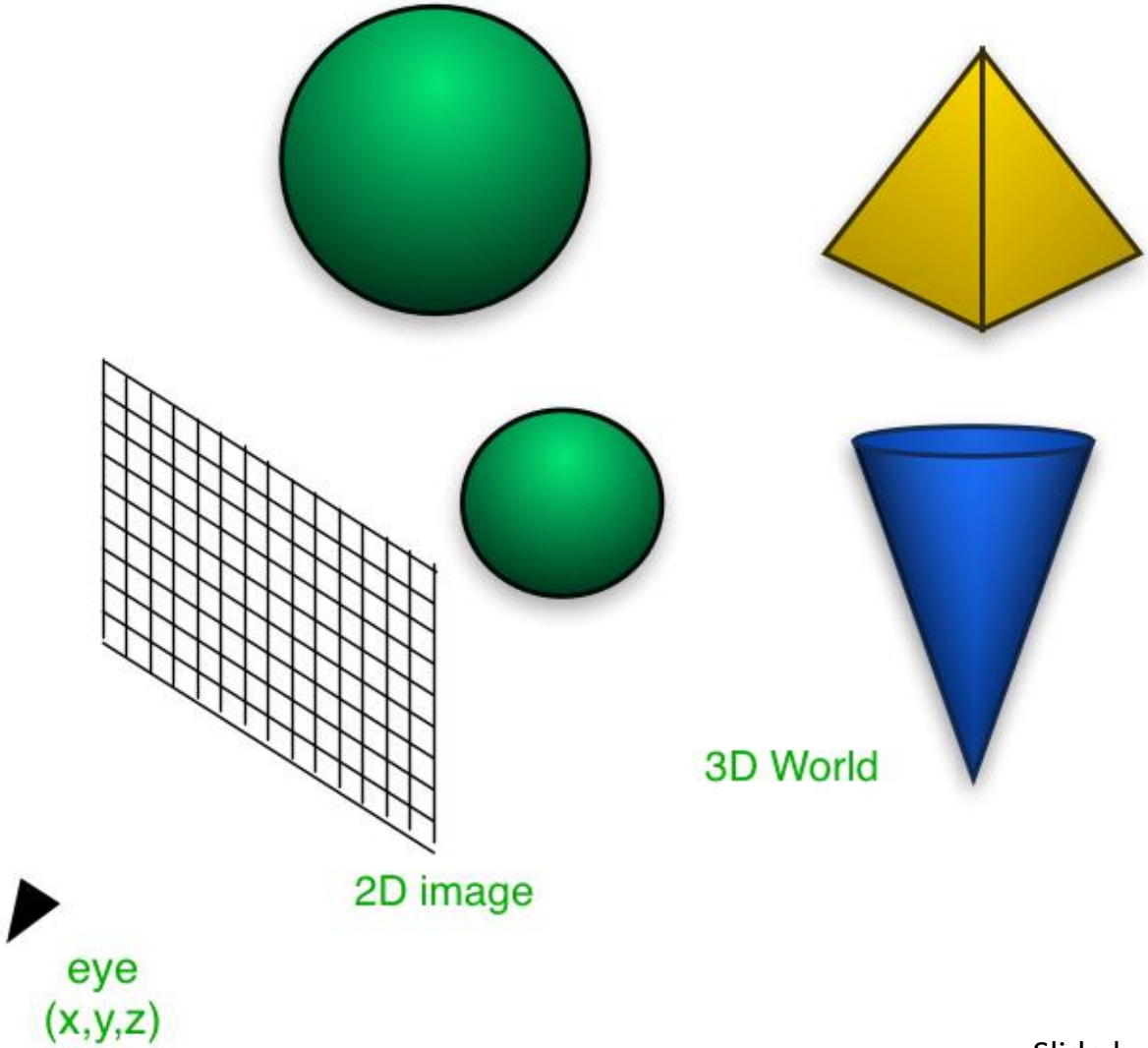






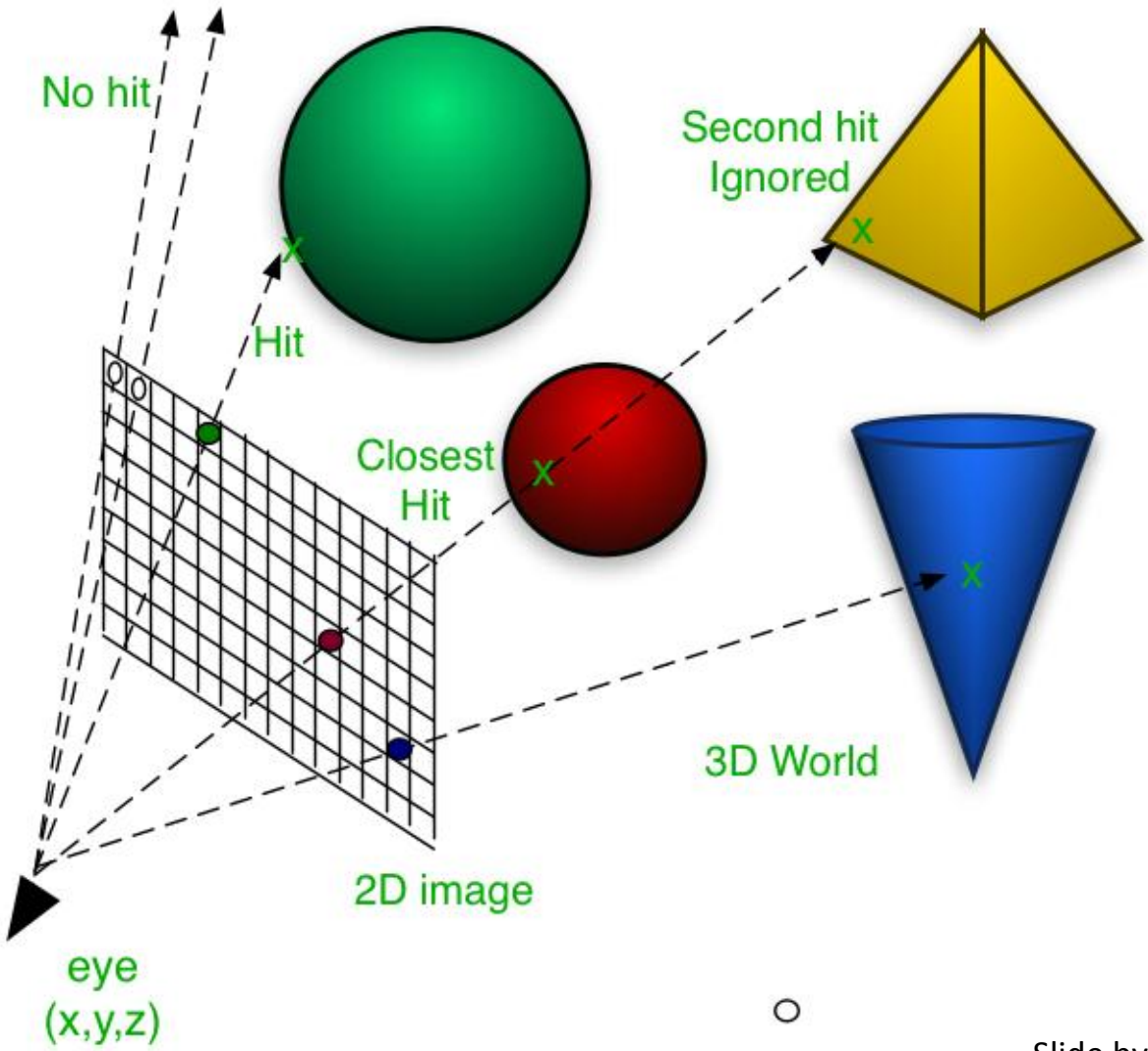
Slide by T. Thorne (U. of Edinburgh)

The input



Slide by M. Blanchette (McGill)

Ray-tracing



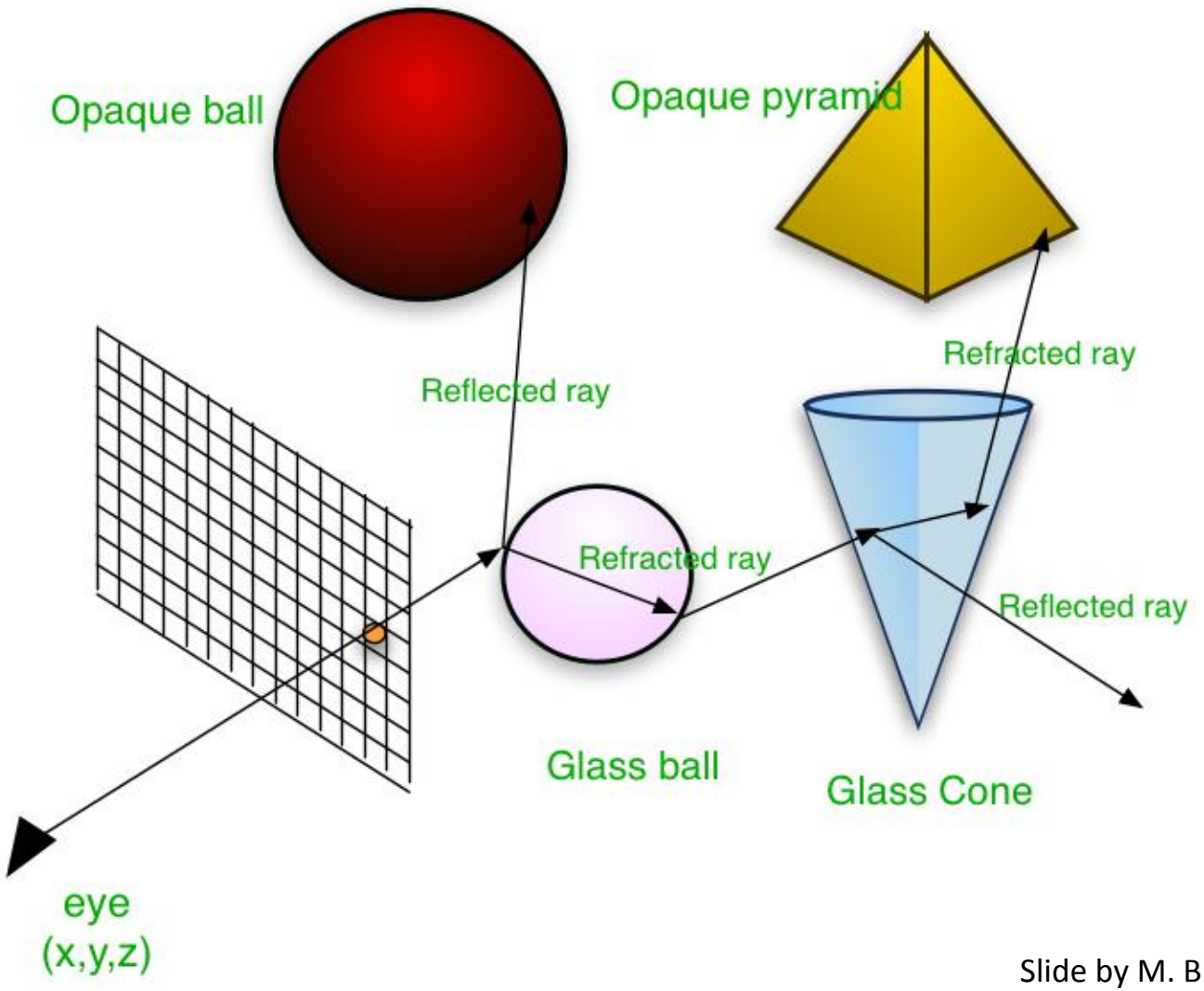
Slide by M. Blanchette (McGill)

Ray-tracing algorithm

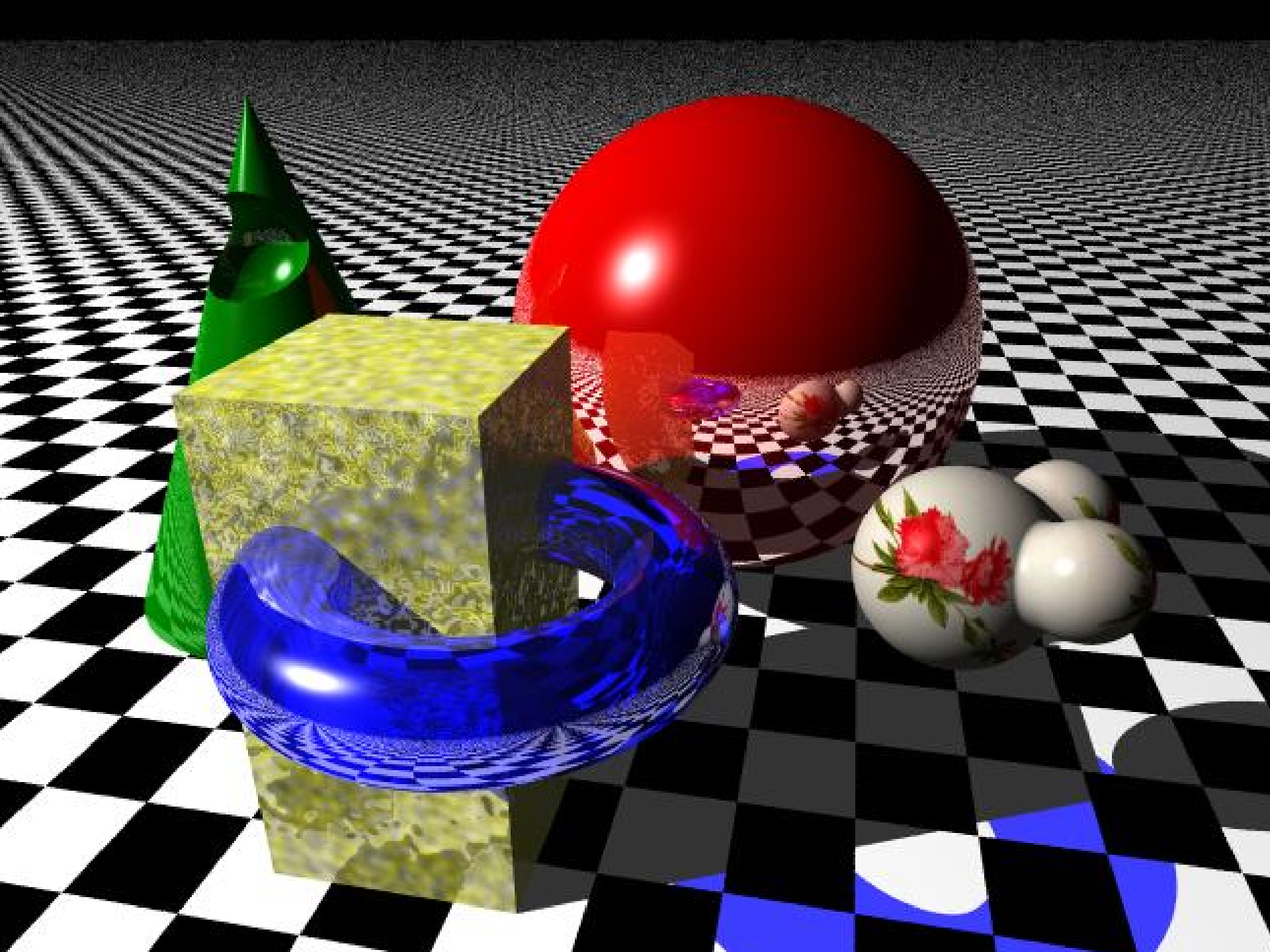
```
Input: - world: set of 3D objects
       - (x,y,z) position of the eye
       - Position of the 2D screen
Output: Image: array of colors of size nPixels by mPixels
For i = 1…nPixels
   For j = 1…mPixels
     r = ray( eye -> pixel(i,j) )
     object = getClosestIntersection(r, world)
     if (object!=null) then
          image[I,j] = object.getColor();
```

Slide by M. Blanchette (McGill)

Recursive Ray-tracing



Slide by M. Blanchette (McGill)



Finding intersections

- Suppose your world consists of Millions of objects
- How can you calculate closest intersection quickly?
  - Computing intersection between ray and each object is much too slow
- Idea: Store your objects in a data structure that allows you to quickly discard objects that can't have intersection

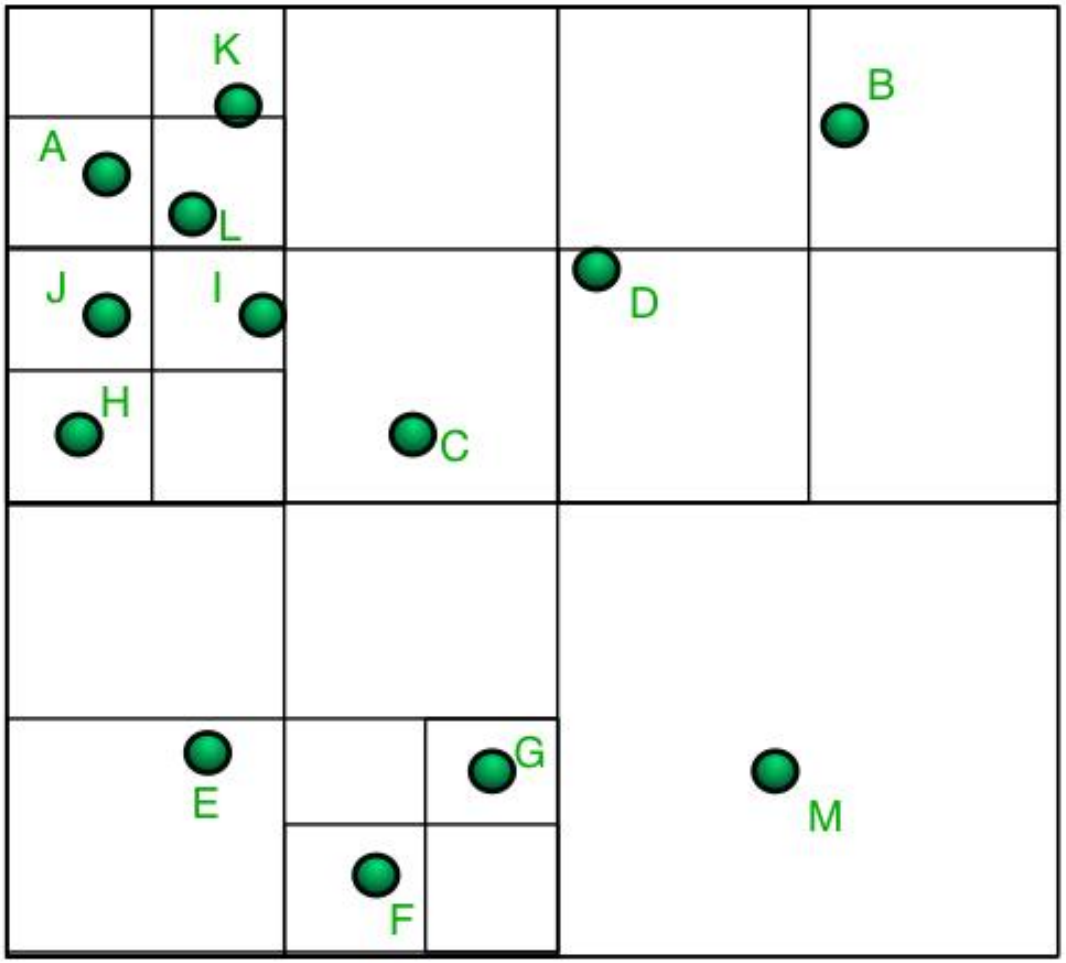
Slide by M. Blanchette (McGill)

Quad trees

For a 2D-world,
Subdivide the world
into four quadrants.

Keep subdividing as long
there is more than one
object per square

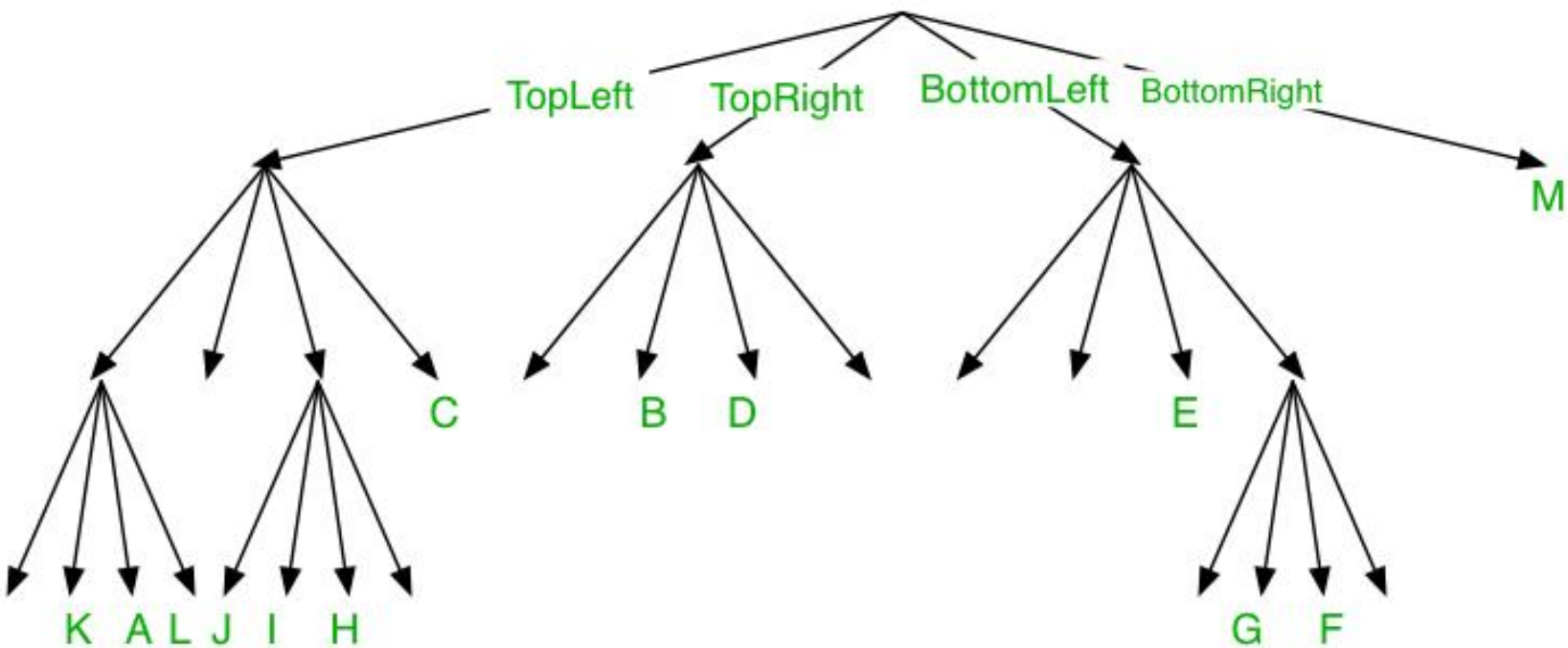
For 3D-world,
Subdivide world into
eight octants



Slide by M. Blanchette (McGill)

Quad trees

Subdivision is represented as a tree:
Root = complete world
Children = four quadrants



Slide by M. Blanchette (McGill)

Fast ray intersection problem

To quickly find intersection between ray and world:

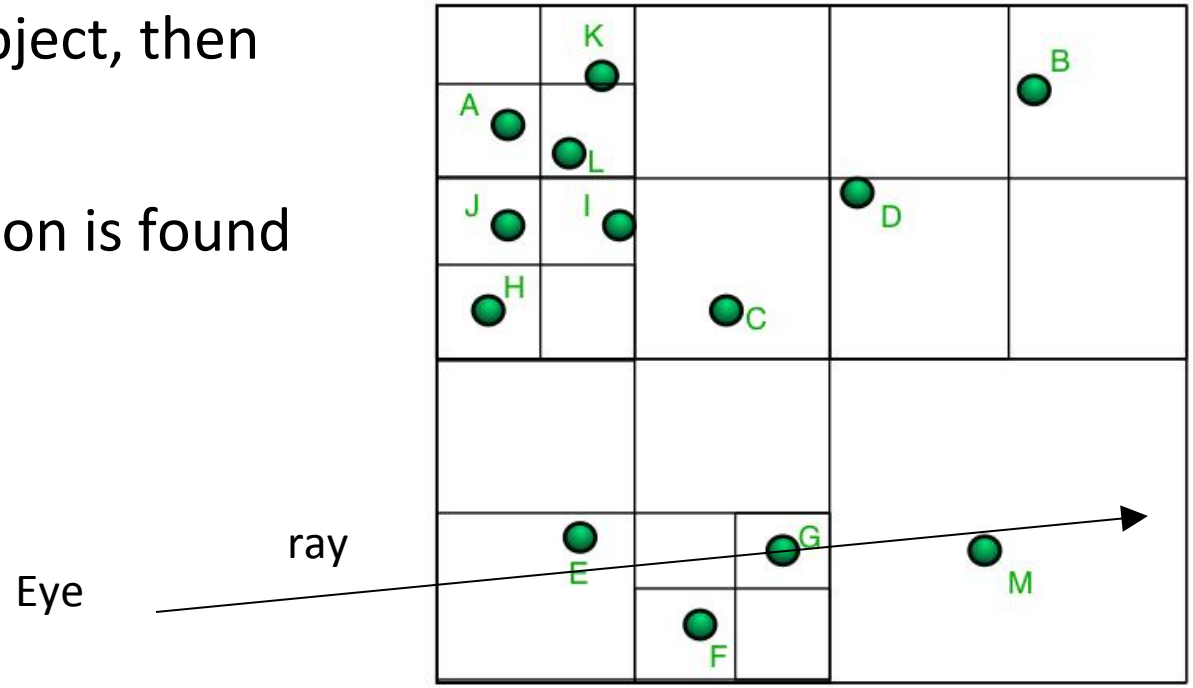
Find which main quadrant is intersected

Find which of its subquadrant is intersected

… Keep going down the tree until a leaf is found

If leaf contains an object, then test intersection

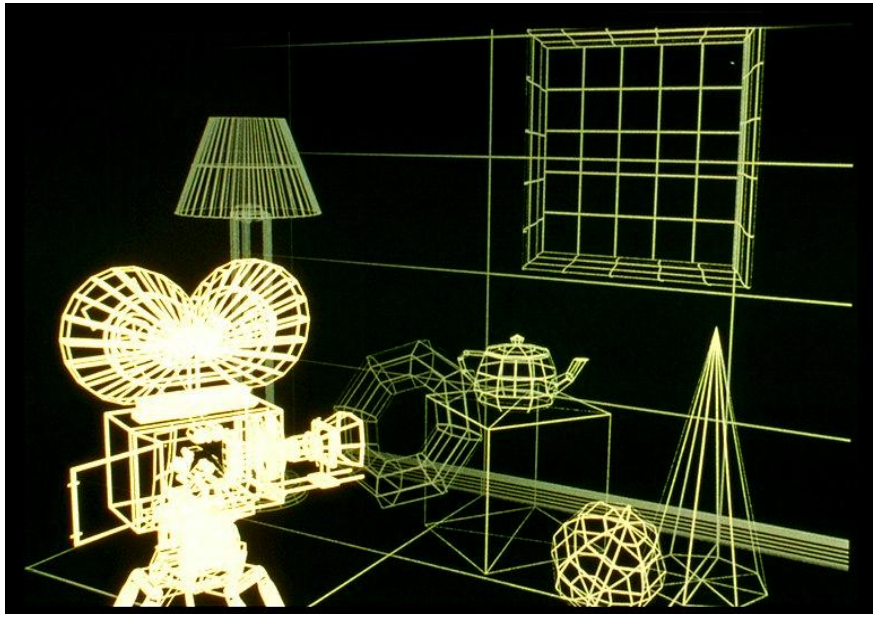
Continue until intersection is found

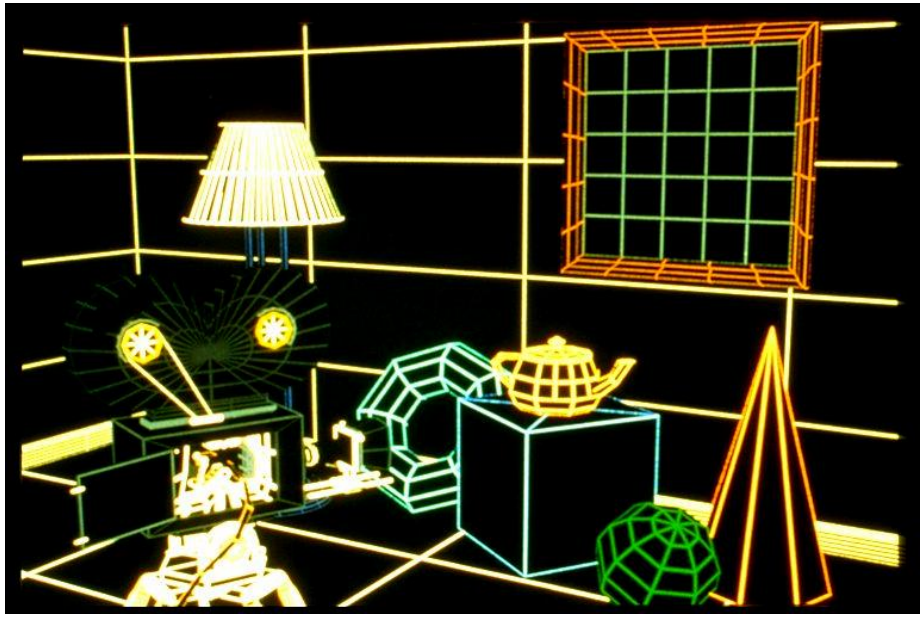


Slide by M. Blanchette (McGill)

Hidden Surface Removal

Objects occluded by other objects must not be drawn

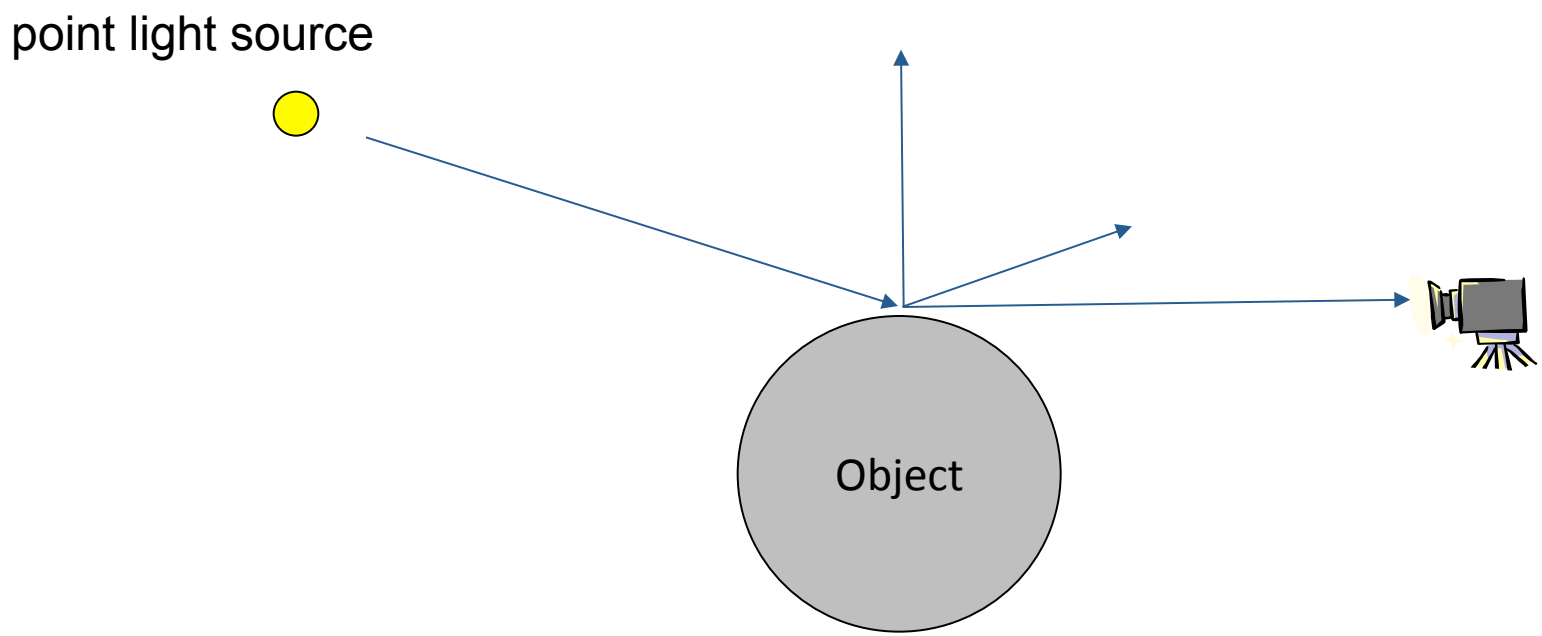




Slide by T. Thorne (U. of Edinburgh)

Shading

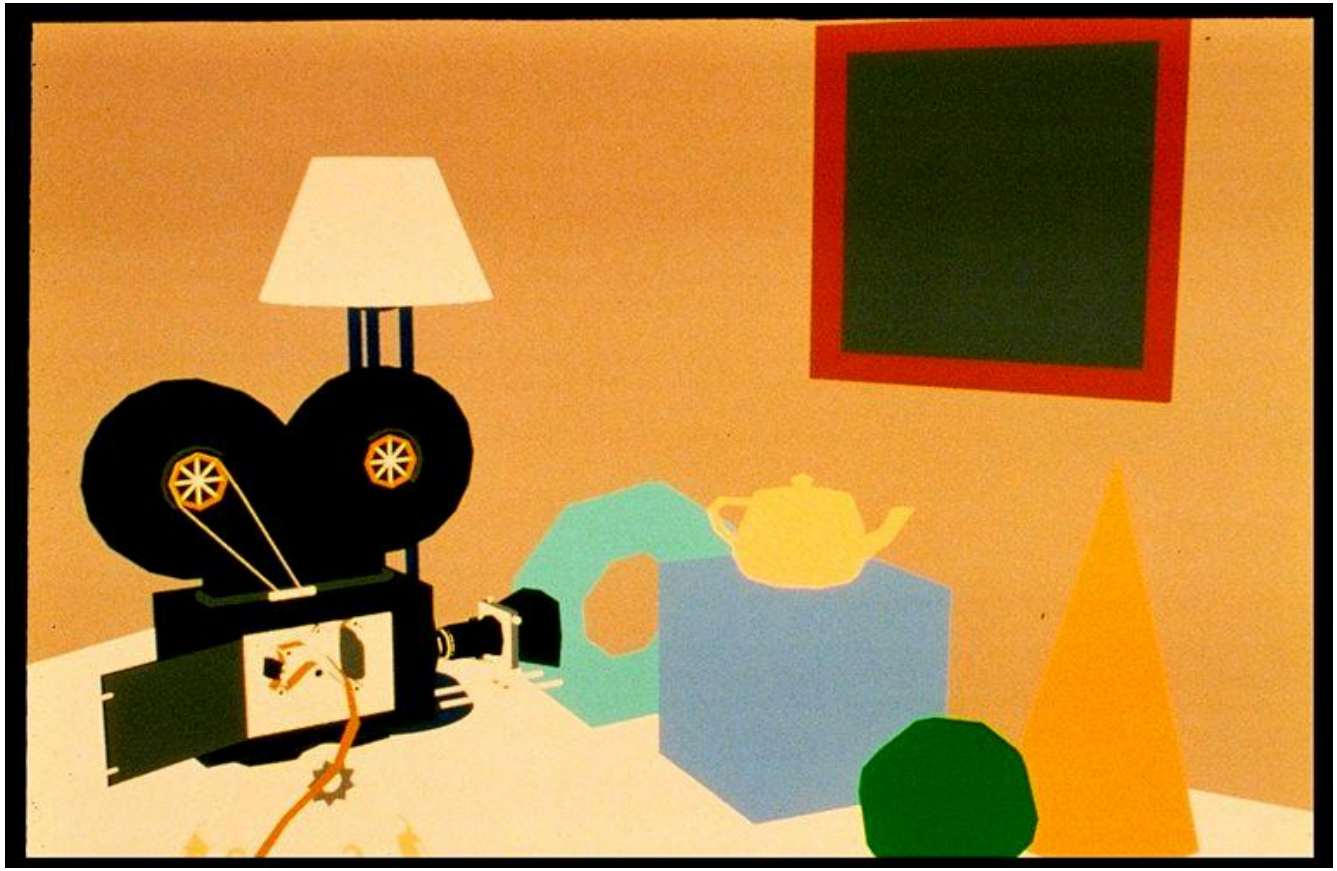
Now we need to decide the colour of each pixels taking into account the object's colour, lighting condition and the camera position



Slide by T. Thorne (U. of Edinburgh)

Shading : Constant Shading - Ambient

Objects colours by its own colour

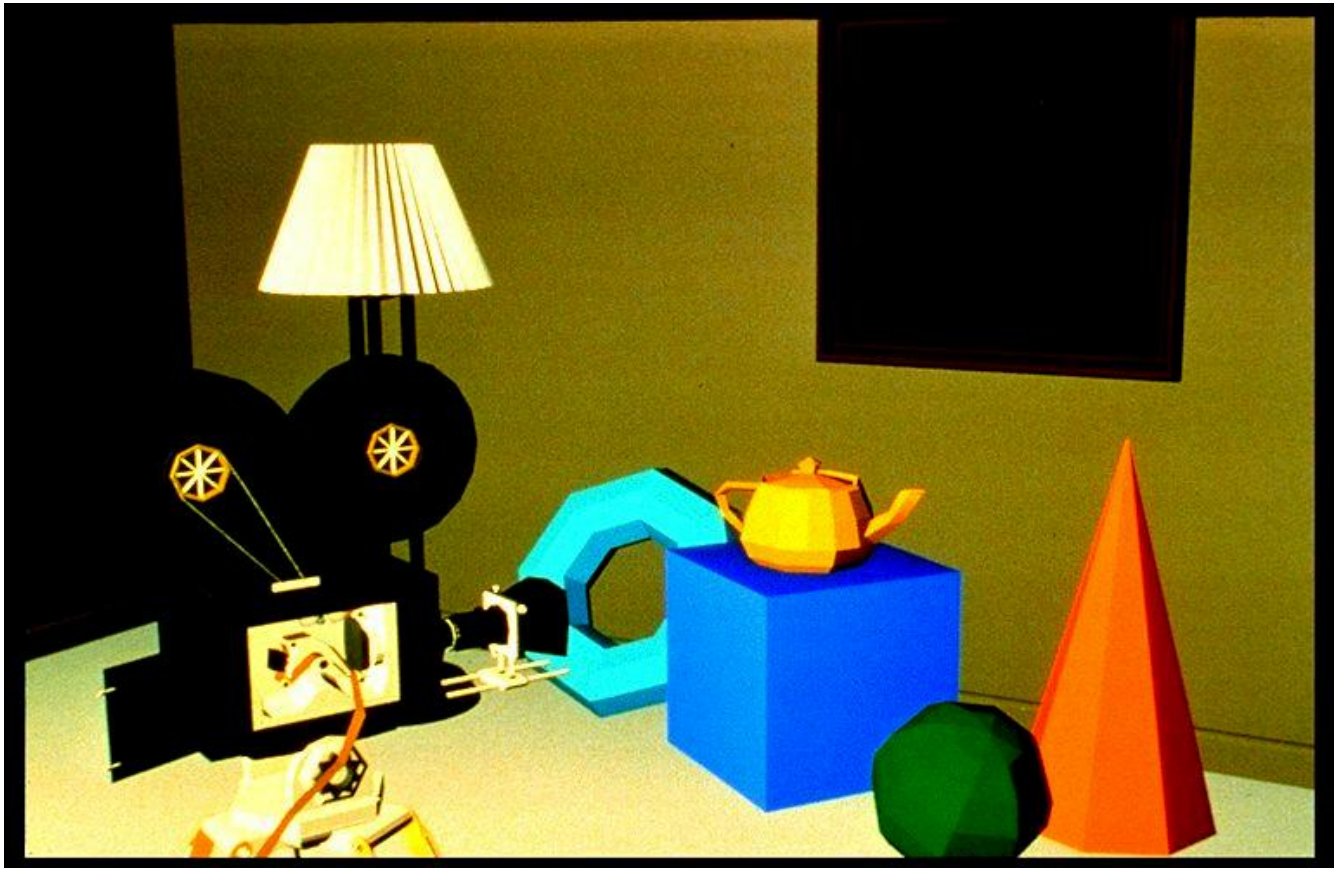


Slide by T. Thorne (U. of Edinburgh)

Shading – Flat Shading

Objects coloured based on its own colour and the lighting condition

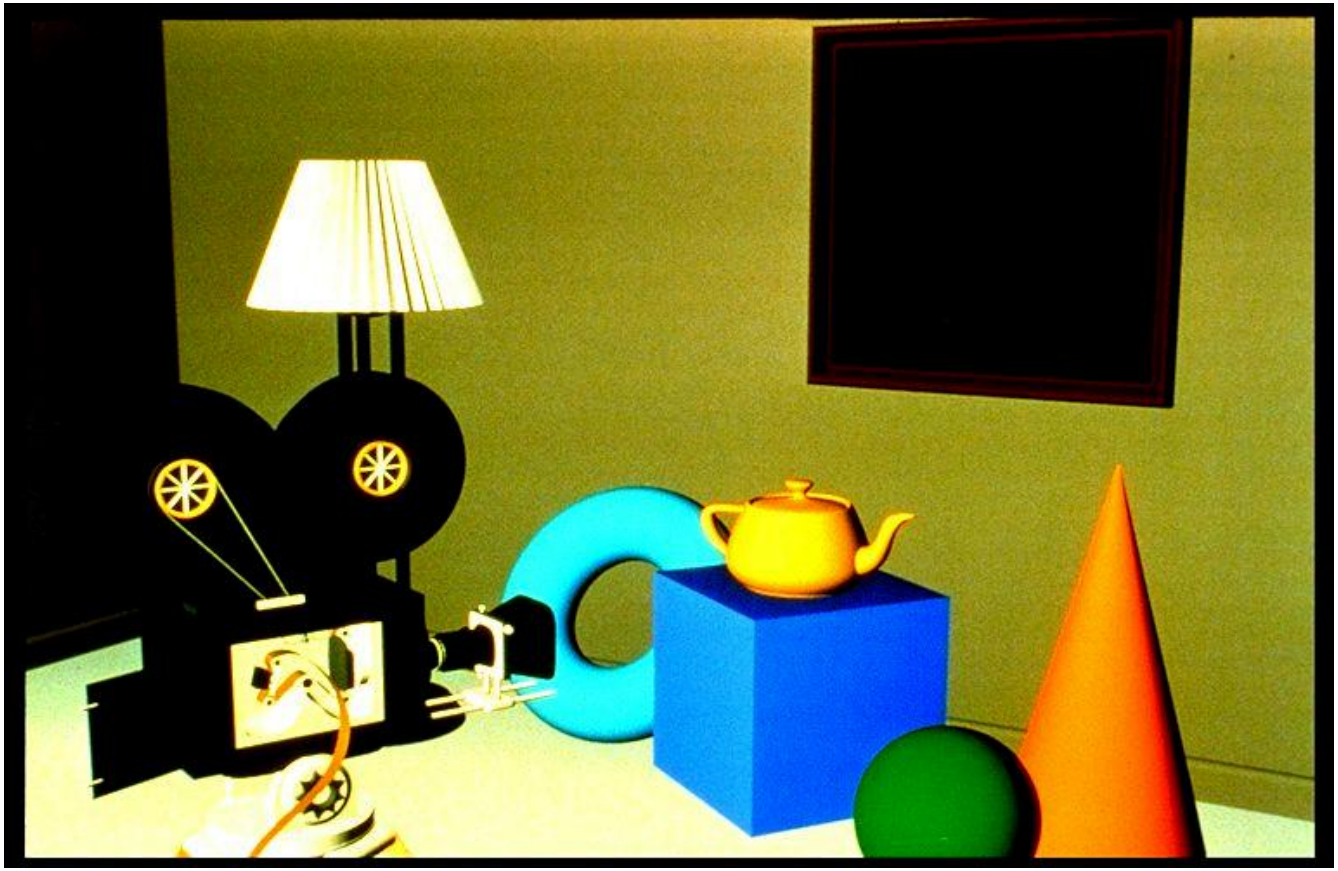
One colour for one face



Slide by T. Thorne (U. of Edinburgh)

Gouraud shading, no specular highlights

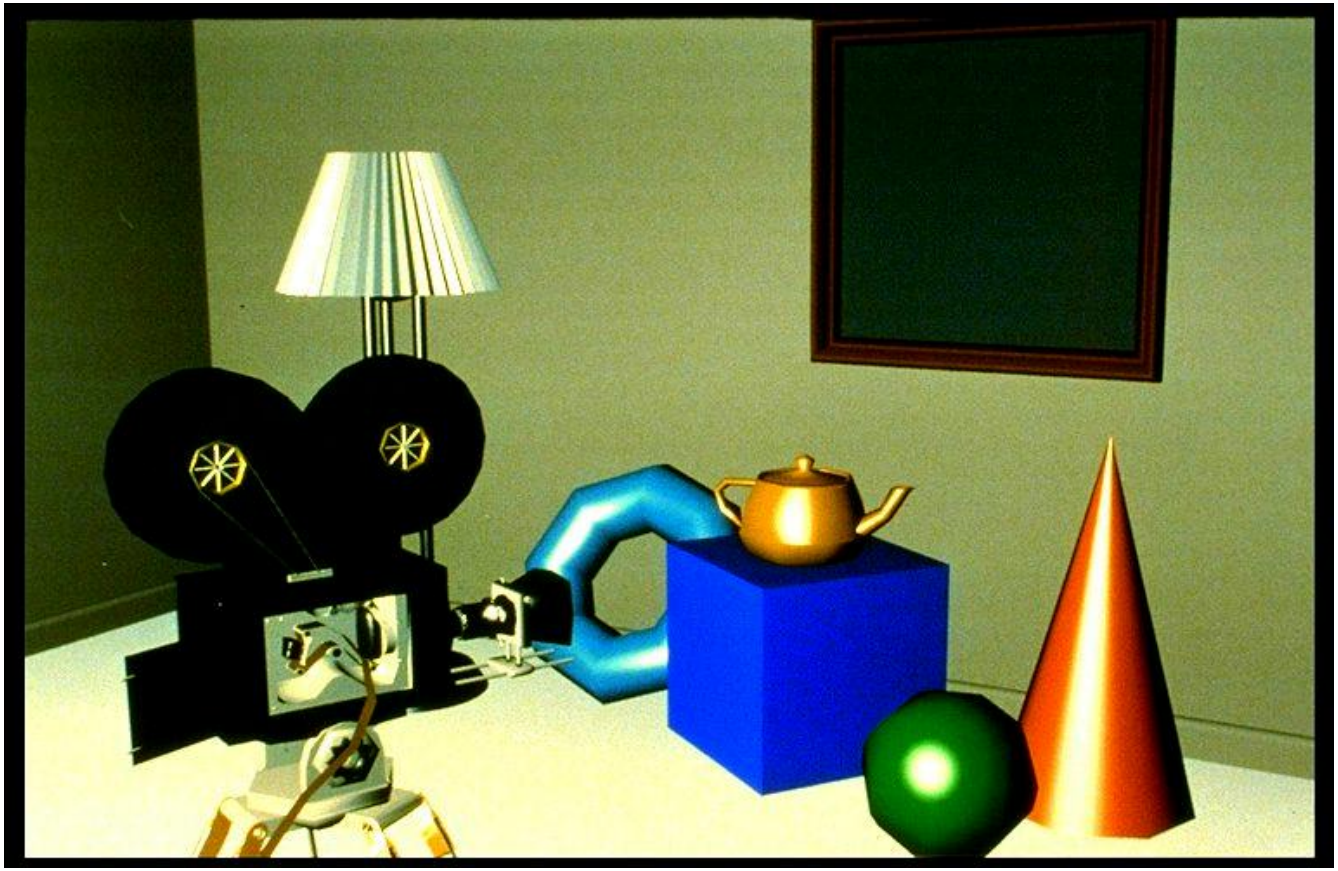
Lighting calculation per vertex



Slide by T. Thorne (U. of Edinburgh)

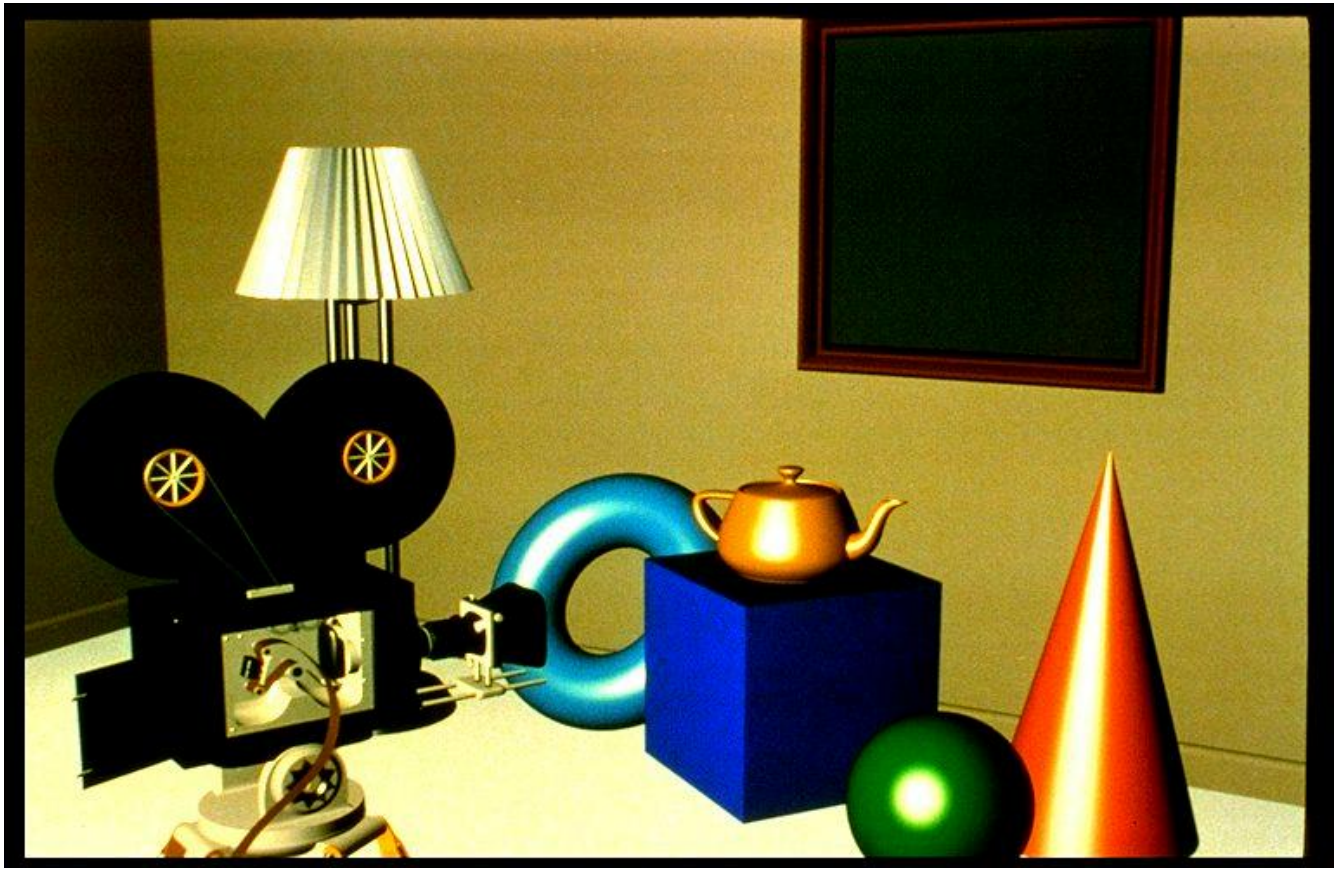
Specular highlights added

Light perfectly reflected in a mirror-like way



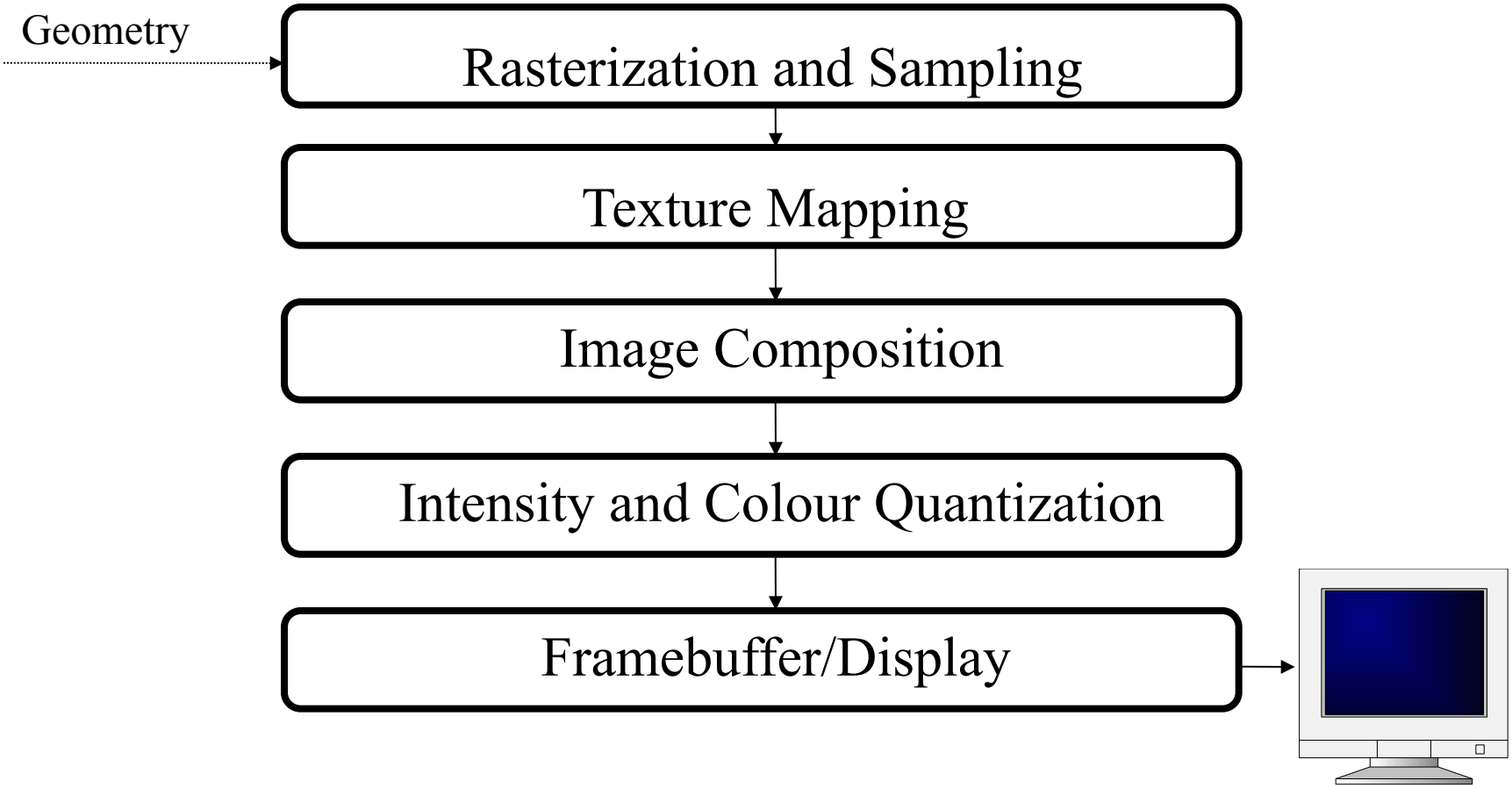
Slide by T. Thorne (U. of Edinburgh)

Phong shading



Slide by T. Thorne (U. of Edinburgh)

Next, the Imaging Pipeline



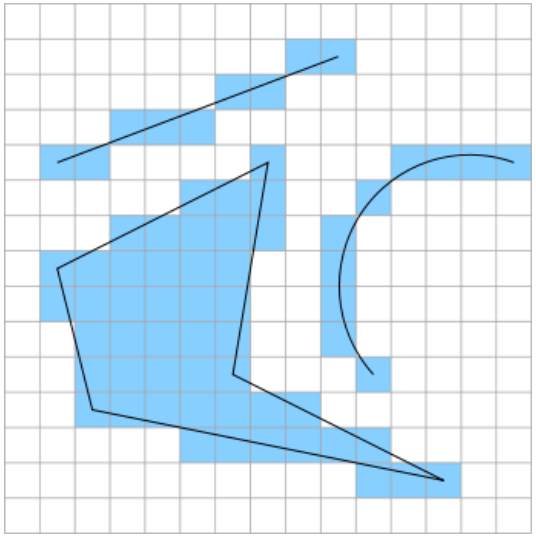
Slide by T. Thorne (U. of Edinburgh)

Rasterization

Converts the vertex information output by the geometry pipeline into pixel information needed by the video display

Aliasing: distortion artifacts produced when representing a high-resolution signal at a lower resolution.

Anti-aliasing : technique to remove aliasing



Slide by T. Thorne (U. of Edinburgh)

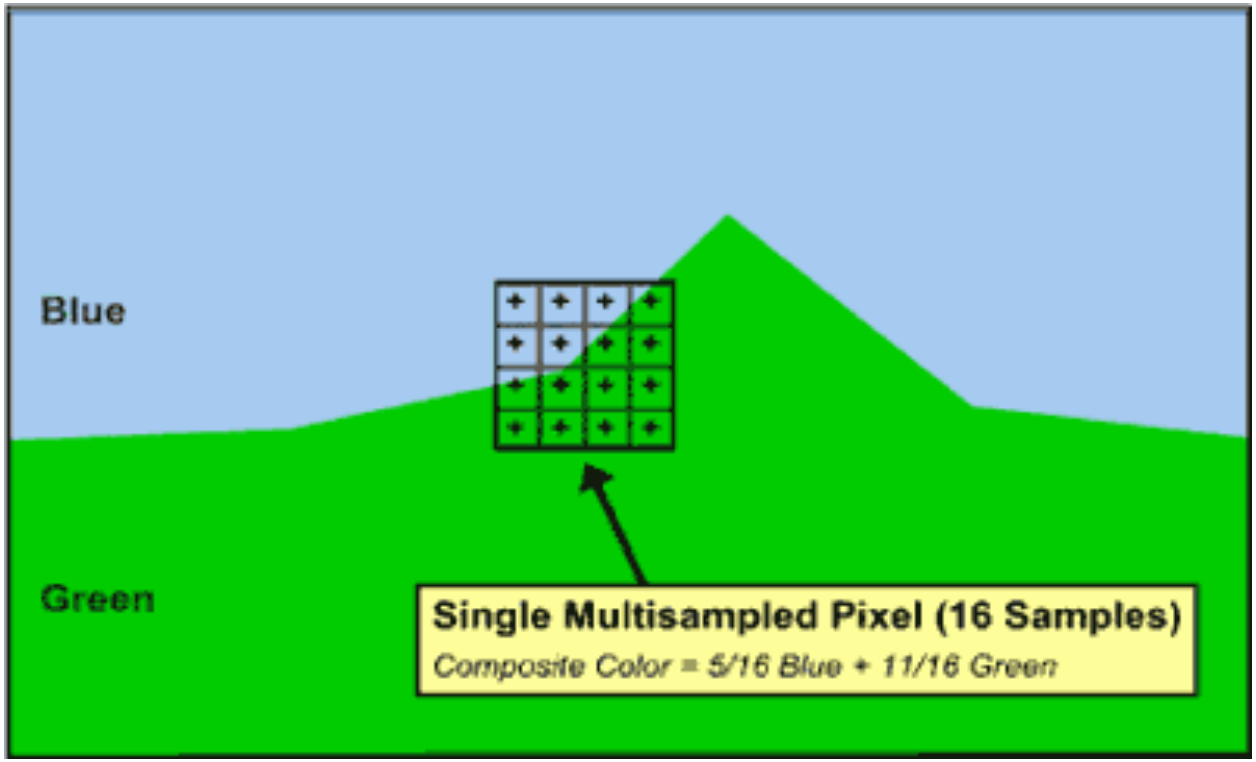
Anti-aliasing

Aliased polygons
(jagged edges)

Anti-aliased polygons

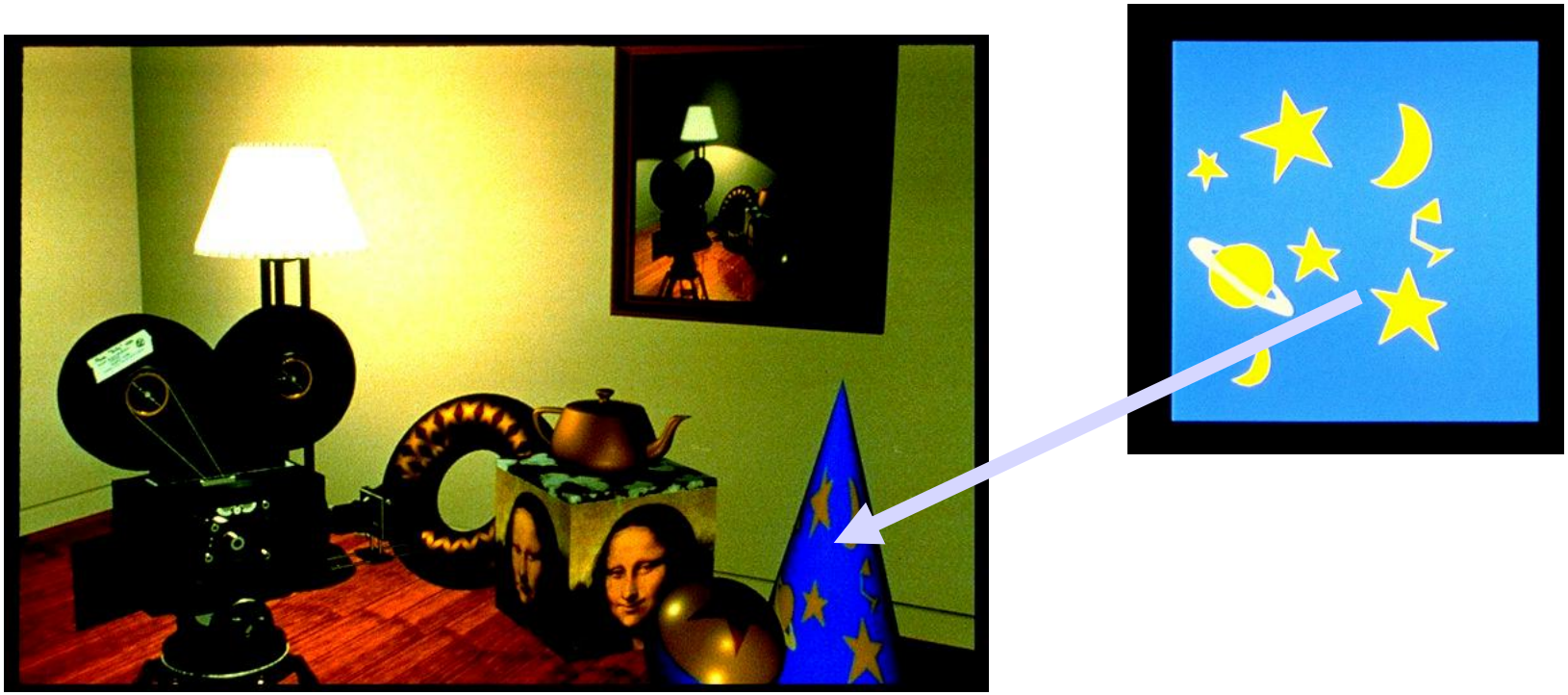
Slide by T. Thorne (U. of Edinburgh)

✓How is *anti-aliasing* done? Each pixel is subdivided (sub-sampled) in n regions, and each sub-pixel has a color;
✓Compute the average color value



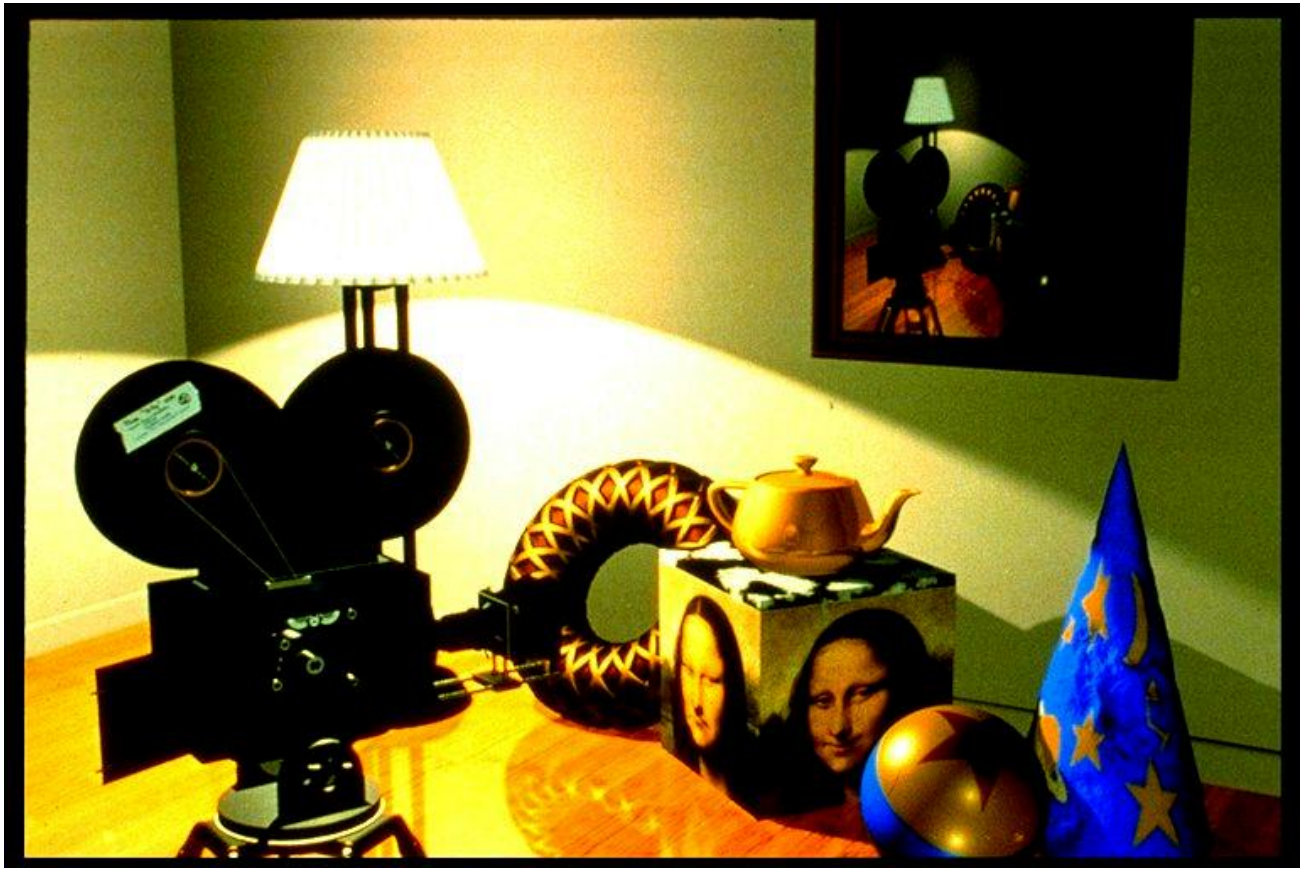
Slide by T. Thorne (U. of Edinburgh)

Texture mapping



Slide by T. Thorne (U. of Edinburgh)

Other topics:
Reflections, shadows & Bump mapping



Slide by T. Thorne (U. of Edinburgh)