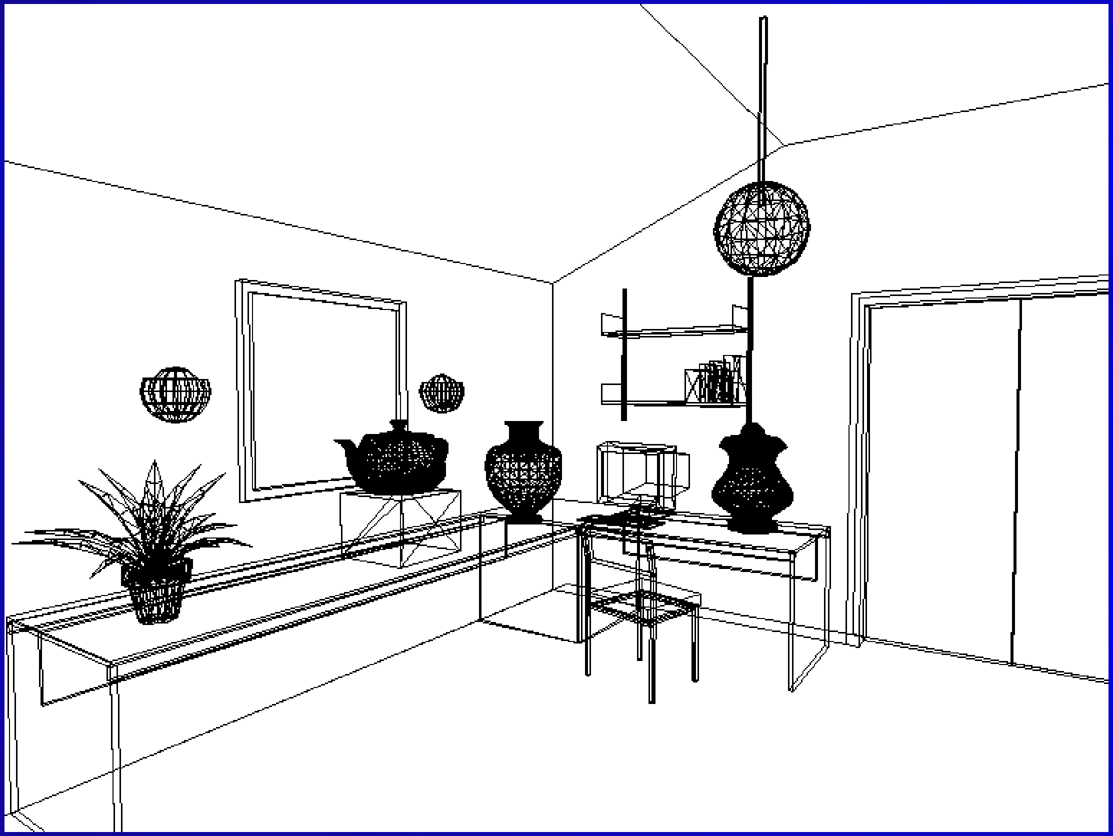
Computer Graphics

Illumination II – Global Models

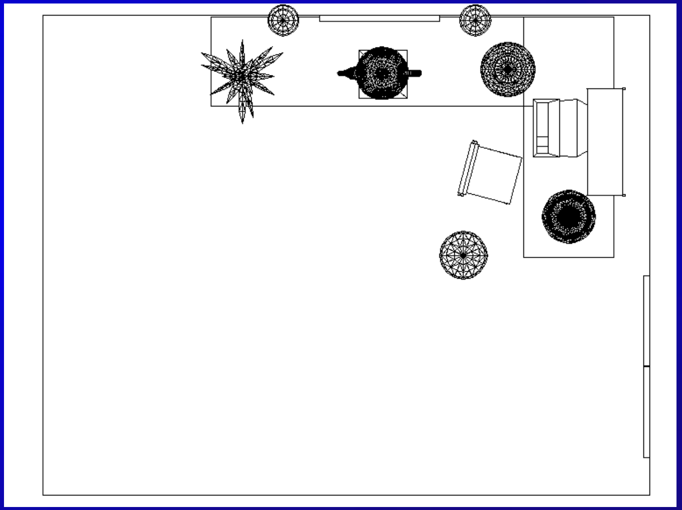
Global Illumination

- Extends the Local Illumination Model to include:
  - Reflection (one object in another)
  - Refraction (Snell’s Law)
  - Transparency (better model)
  - Shadows (at point, check each light source)
  - Antialiasing (usually means supersampling)

Wireframe view of a test scene.



Orthographic view from above



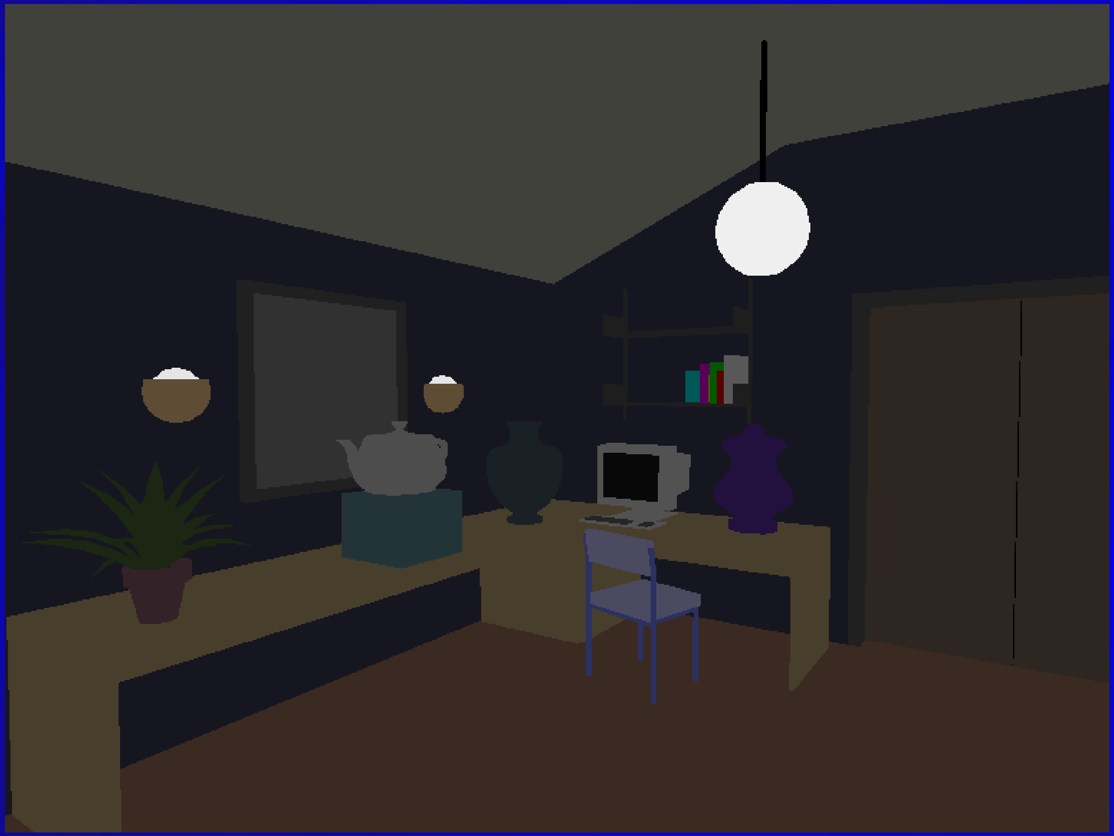
Test Scene.

High quality rendering of test scene.

Note :

- Mirror and chrome teapot.
- Shadows on floor.
- Shiny floor.

Locally illuminated test scene.



Ambient term only

Locally illuminated test scene.

Phong shading.

Ambient and Diffuse terms only

Notes :

- Highlight on wall from light is in the wrong place; screen space interpolation.
- We cannot illuminate the lights with the light sources – wrong side !

Locally illuminated test scene.

Phong shading.

Ambient, diffuse and Specular terms.

Locally illuminated test scene.

Flat shading.

Note : Mach bands.

Locally illuminated test scene.

Gouraud shading.

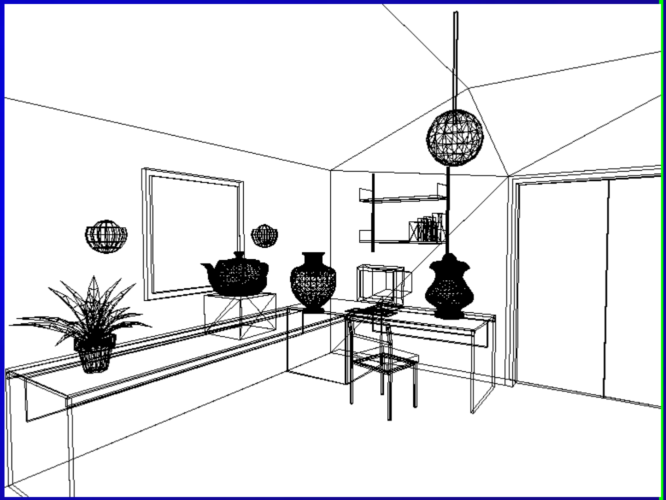
Ambient, diffuse and Specular terms.

Note: artefacts on wall.

Solution to Gouraud artefacts.

Gouraud shading.

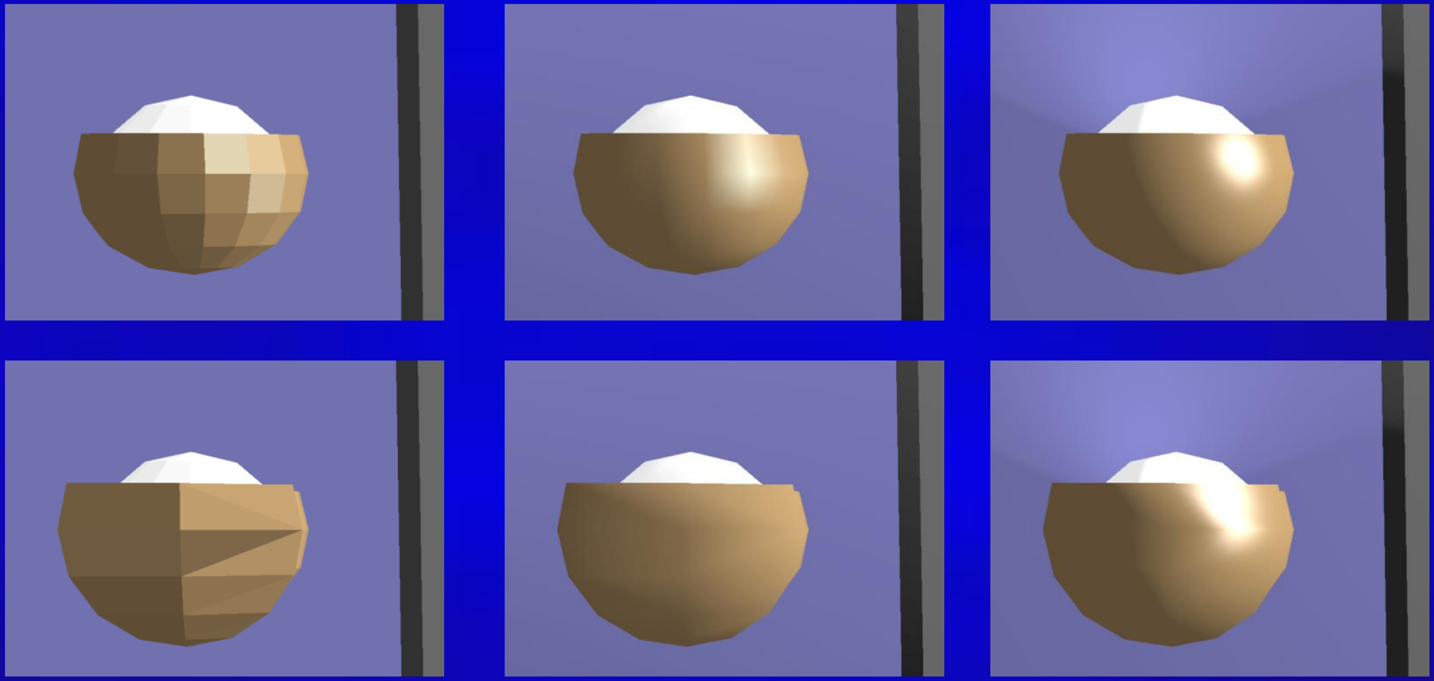
Re-triangulated mesh.



Comparison.

Flat Gouraud Phong

Coarser mesh



Use Local illumination.

- No.
- In our test scene, we can't represent :
  - Mirror
  - Chrome teapot.
  - Shiny floor
  - Shadows

with local illumination.

Global illumination.

- Two methods :
  - View dependent methods.
    - Calculate the view from the camera with global illumination.
  - Recursive ray-tracing.

  - View independent methods.
    - Solve lighting for the entire scene.
    - Radiosity solution.

View dependent methods.

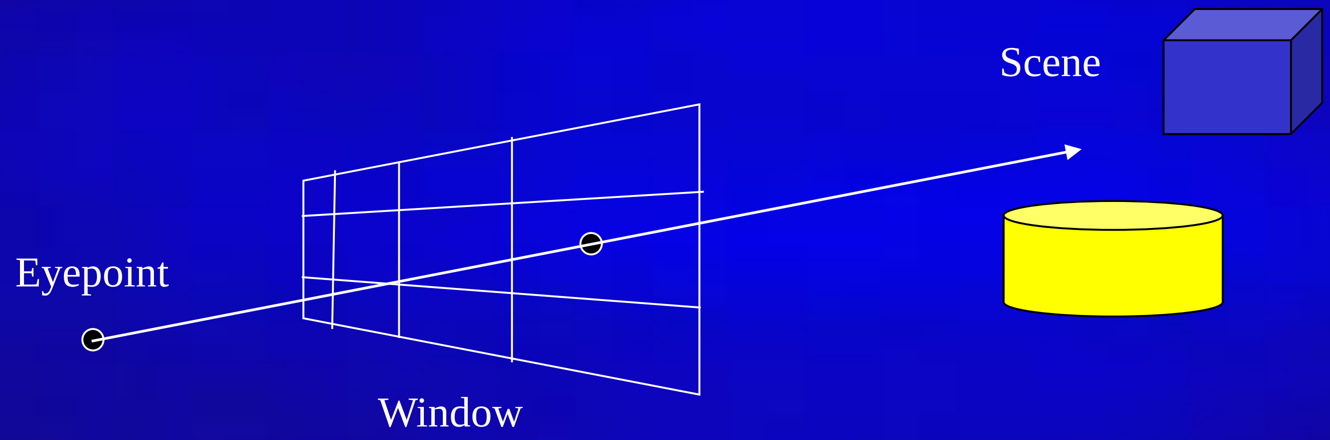
- Loop round the pixels…..
- Good for lighting effects which have a strong dependence on view location :
  - Specular highlights.
  - Reflections from curved surfaces.
- Only a small number of objects need to be considered at the same time.
- Poor when many objects need to be considered
  - E.g diffuse interactions (eg. colour bleeding).

View independent methods.

- Loop round the scene…
- Good when many (all) objects need to be considered at same time.
  - Diffuse inter-reflections.
- Poor when shading has strong dependence on view location.
  - Specular reflection.

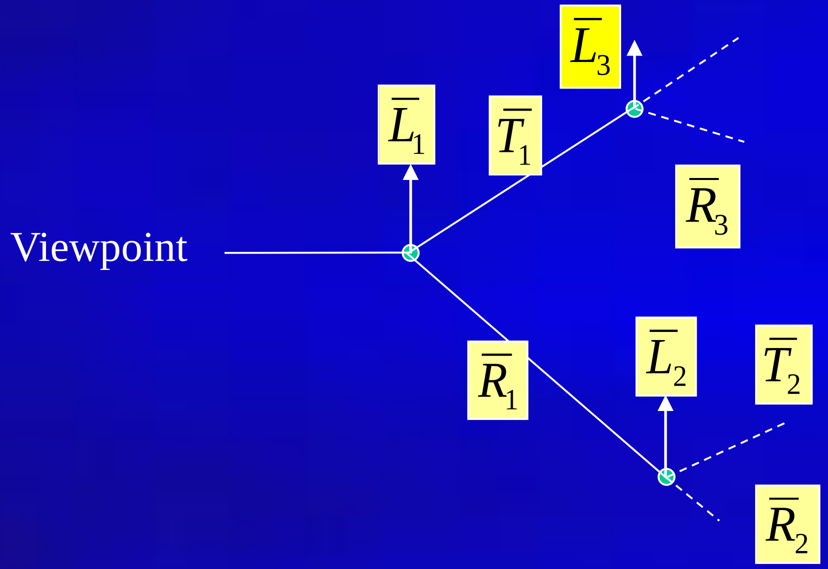
Recall : Ray casting.

- Involves projecting an imaginary ray from the centre of projection (the viewers eye) through the centre of each pixel into the scene.
- The first object the ray intersects determines the shade.



Recursive ray tree.

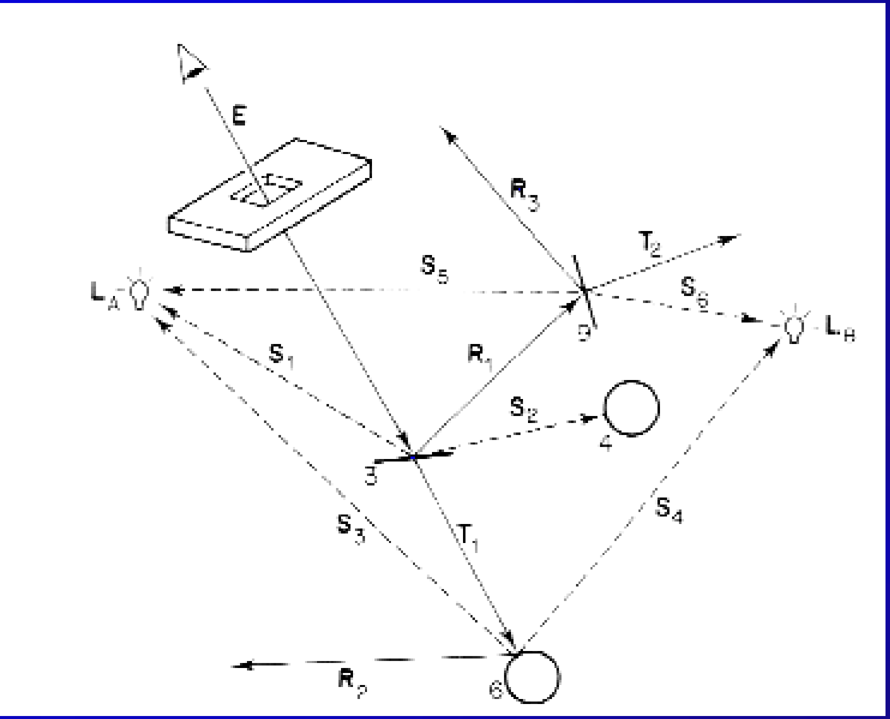
- Reflection and Transmission Rays spawn other rays.
  - Shadow rays test only for occlusion.
- The complete set of rays is called a *Ray Tree*.



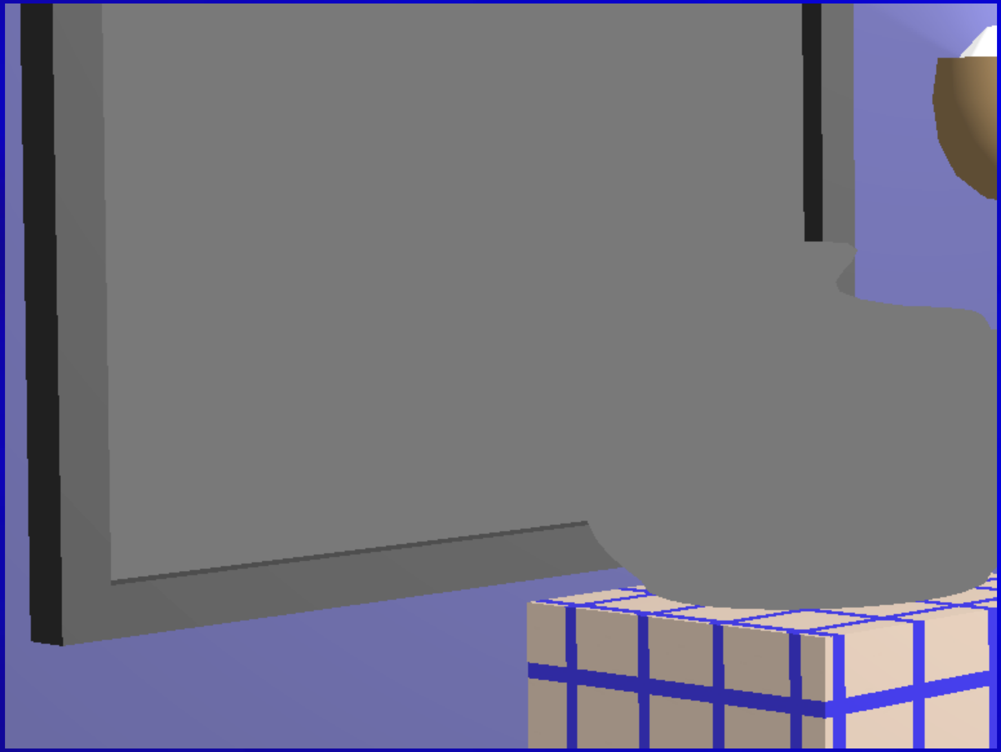
Light Source ray determines colour of current object.

Recursive ray tree.

- Reflection and Transmission Rays spawn other rays.
  - Shadow rays test only for occlusion.
- The complete set of rays is called a *Ray Tree*.



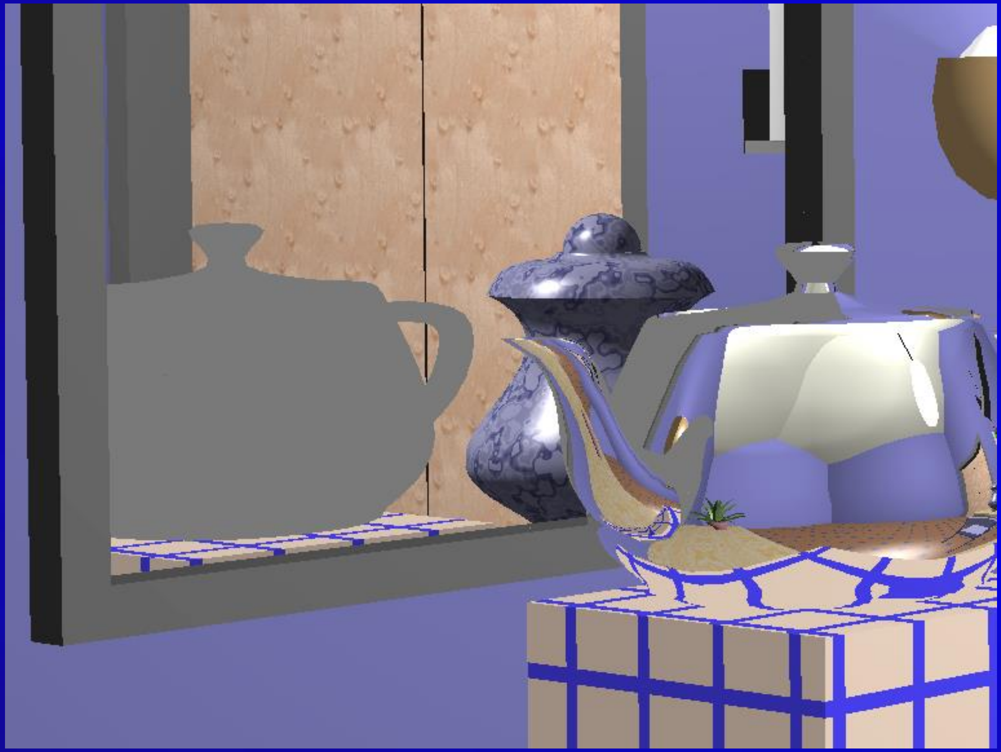
Test Scene.



Ray tree depth 1.

Note only ambient shade on mirror and teapot

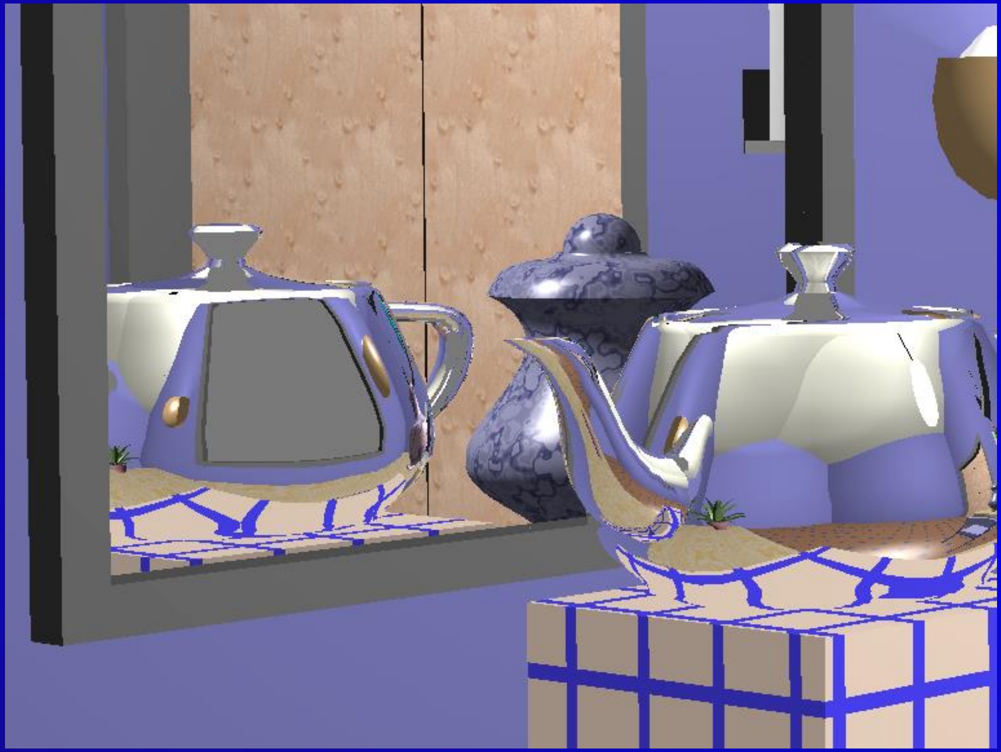
Test Scene.



Ray tree depth 2.

Note only ambient shade on reflection of mirror and teapot.

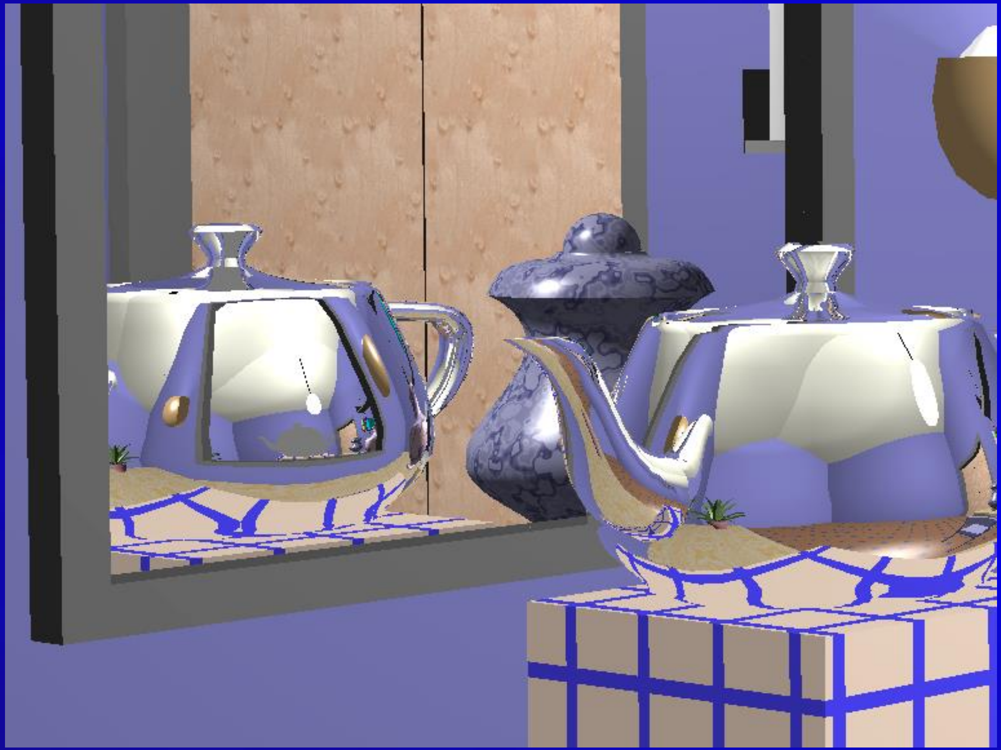
Test Scene.



Ray tree depth 3.

Note only ambient shade on reflection of mirror in teapot.

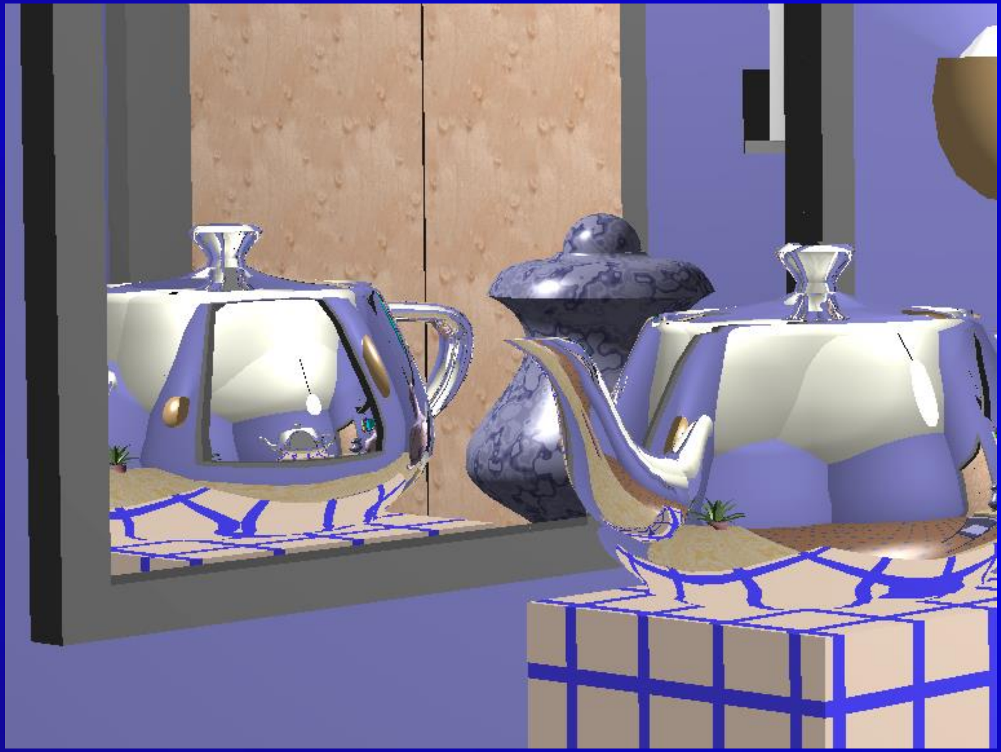
Test Scene.



Ray tree depth 4.

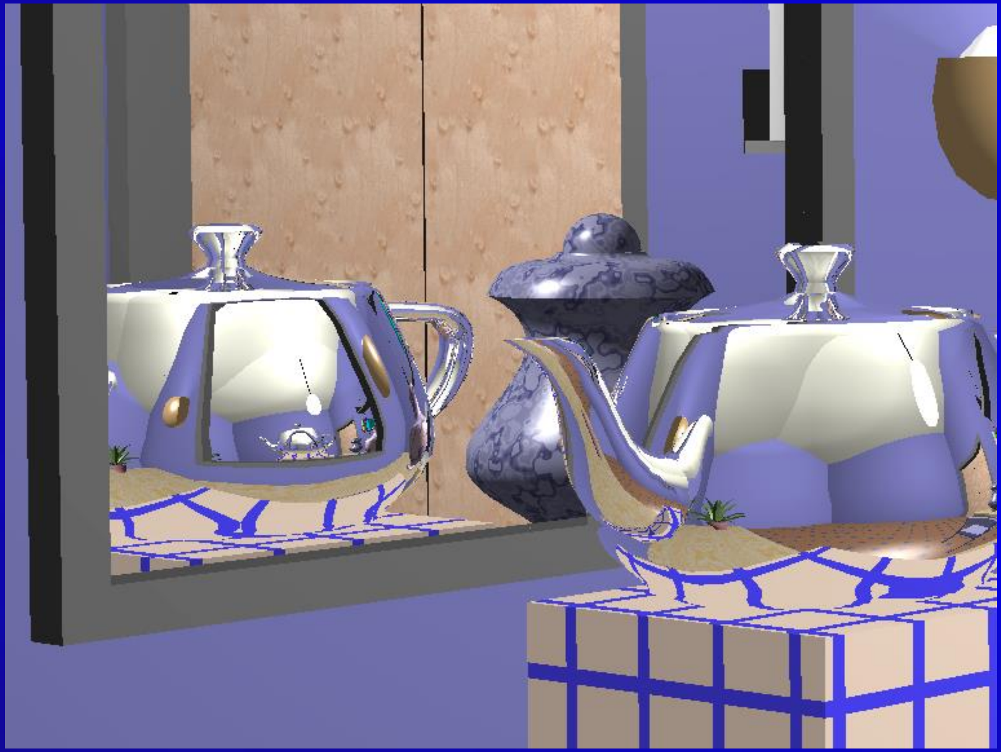
Note ambient shade on reflection of teapot in reflection of mirror in teapot.

Test Scene.



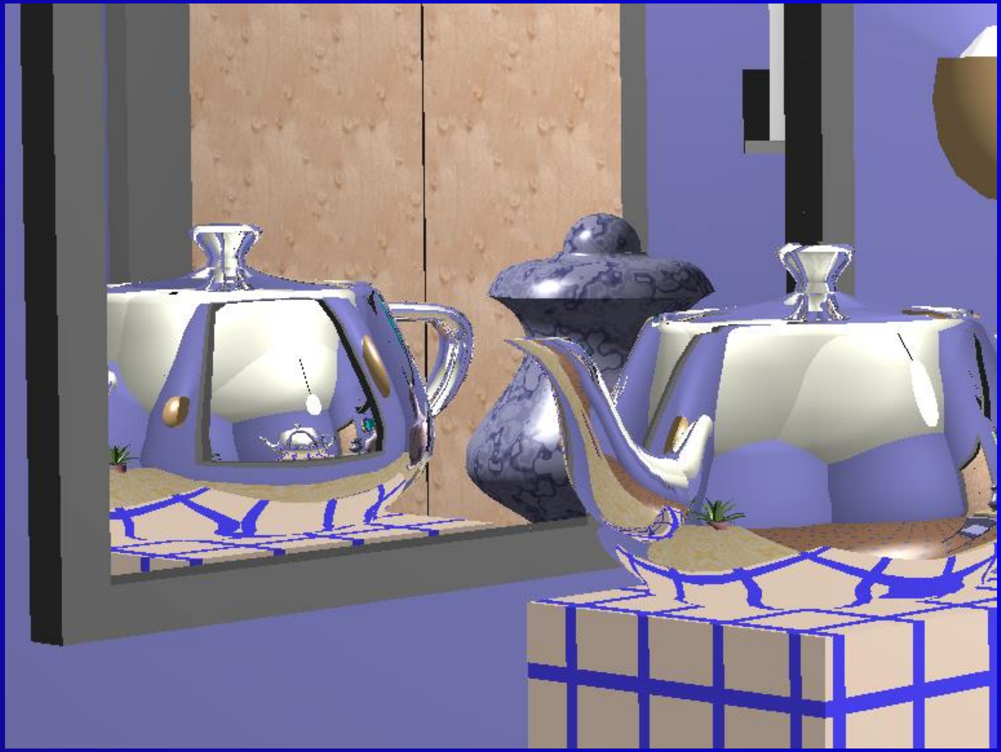
Ray tree depth 5.

Test Scene.



Ray tree depth 6.

Test Scene.

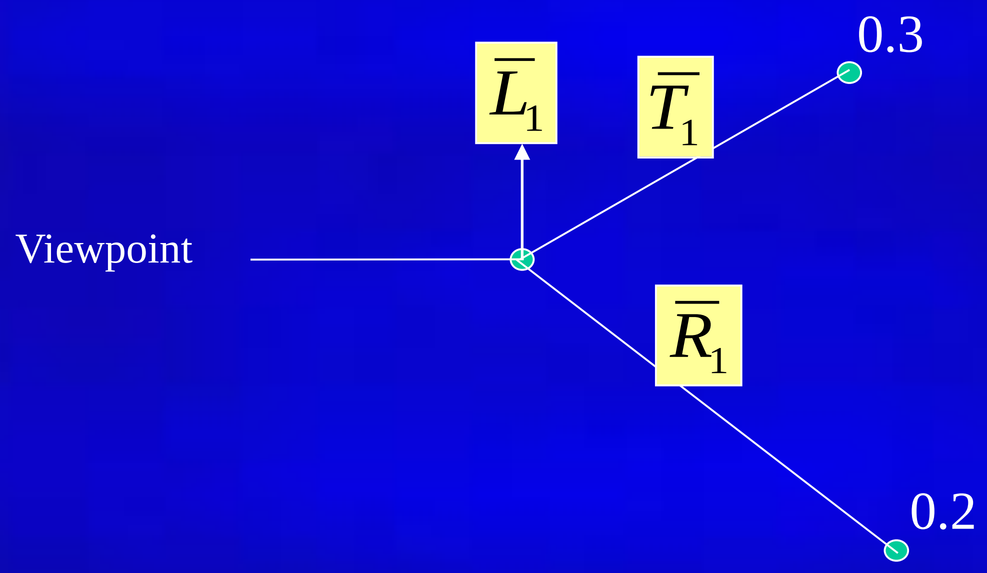


Ray tree depth 7.

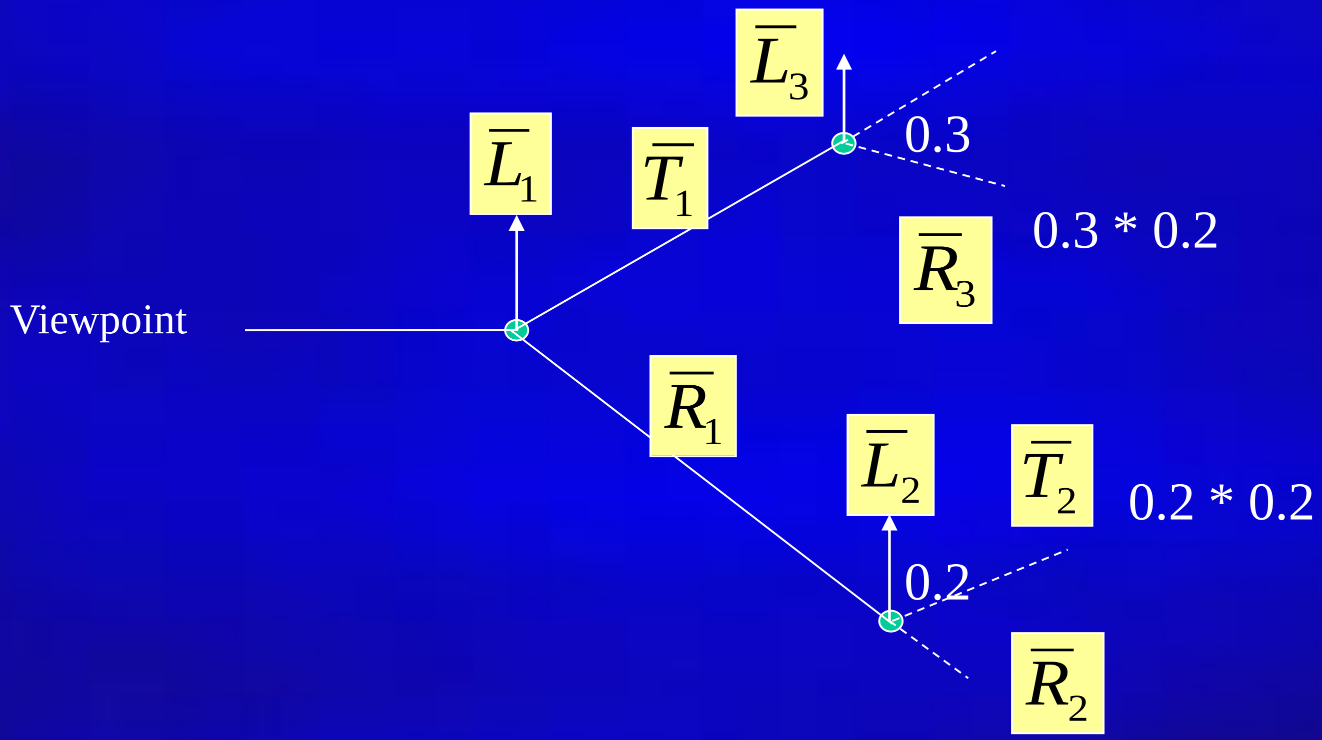
When to stop ?

- Need to know when to stop the recursion.
- Can define a fixed depth.
- Hall introduced *adaptive tree depth control.*
  - Calculate maximum contribution of a ray to a pixels final value.
  - Multiply contribution of ray's ancestors down the tree.
  - Stop when below some threshold, perhaps stack overflow.
  - May miss major contribution this way (culled bright pt)

Adaptive tree depth control.



Adaptive tree depth control.



Global vs. Local illumination.

- In both an object hit by a ray, if lit by a light source, is illuminated by a local illumination model, i.e with specular, diffuse & ambient terms.
- Global: a reflected ray, a shadow feeler, and a transmission ray (if appropriate) are also cast into the scene.
- Phong term only reflects light source.
  - Need to adjust local illumination terms to normalise total light values.
  - Inconsistent if local and global specular terms used together as local term spreads light source, global term does not.

Incorrect result.

- Effect of not normalising reflection and transmission – light appears to be created.
  - Reflection & transmission = 100%

Problems with Ray-tracing.

- A serious problem with Ray tracing is rays are traced from the eye.
  - Refraction is not physically correct.
- Shadow rays are cast only to light sources
  - Lights reflected in mirrors do not cast shadows
  - Shadows of transparent objects don't exhibit refraction.
  - Still need local illumination for diffuse shading.

Speeding up Ray Tracing

- Ray tracing is slow, not real-time.
- Use appropriate extents for objects.
- Ray tracing is inherently parallel.
- Use item buffers – z-ordered lists, store closest object per pixel.
- Use light buffer – z-ordered list per light ray used for shadowing.

Recommended Reading

- Foley at al. Chapter 16, sections 16.11, 16.12 and 16.12.5.
- Introductory text Chapter 14 sections 14.6 and 14.7.
- Most graphics texts cover recursive ray tracing.