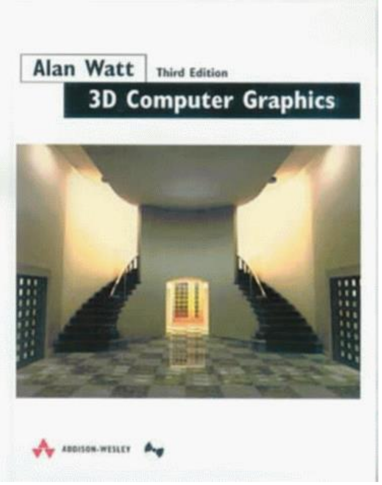
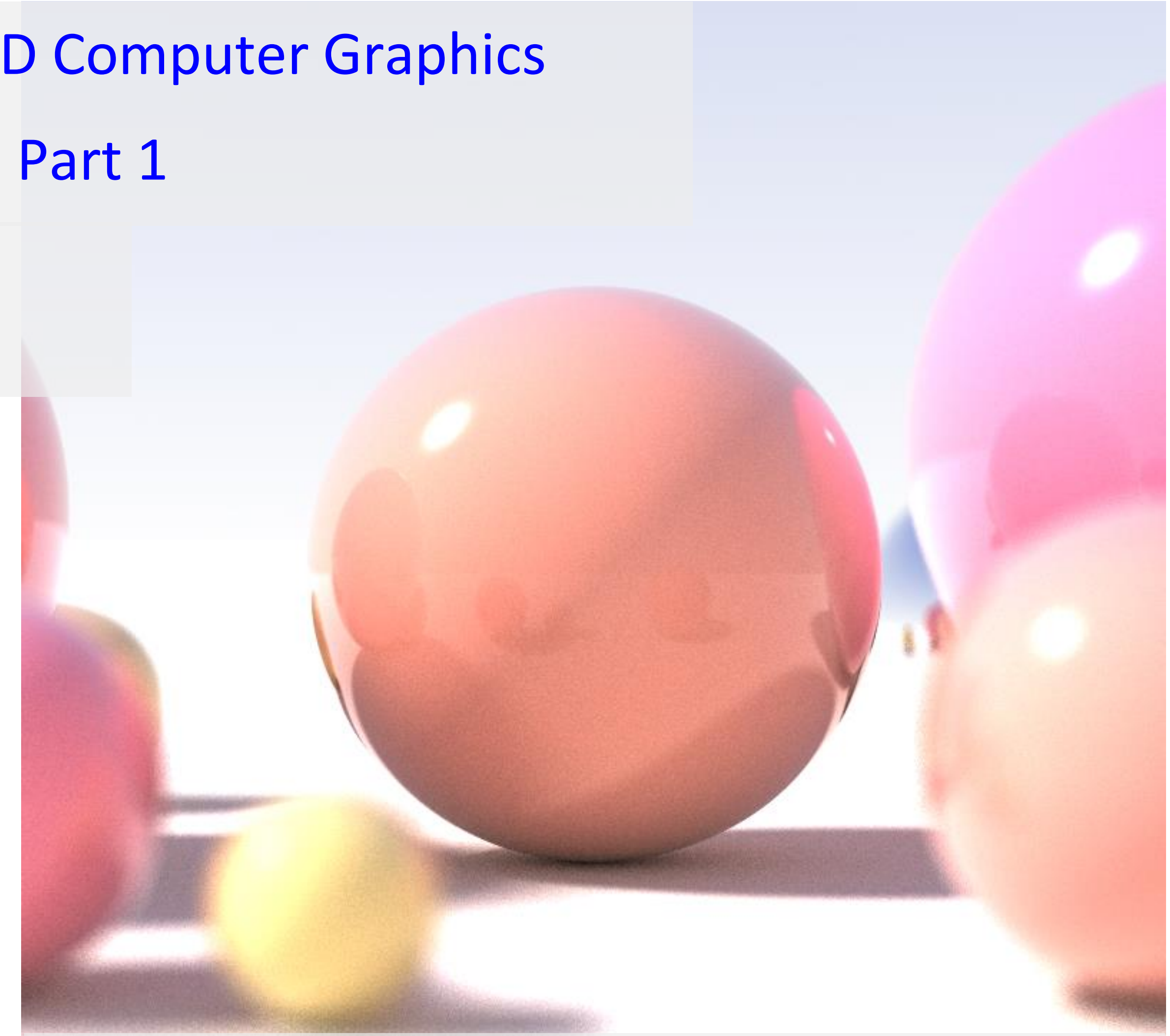
COM3503/4503/6503: 3D Computer Graphics

Lectures 17: Ray tracing: Part 1

Dr. Steve Maddock
s.maddock@sheffield.ac.uk

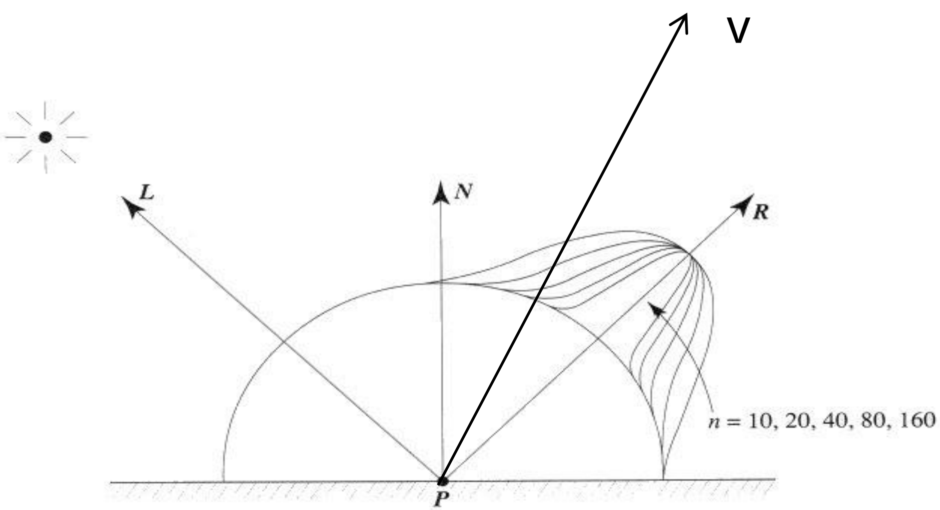


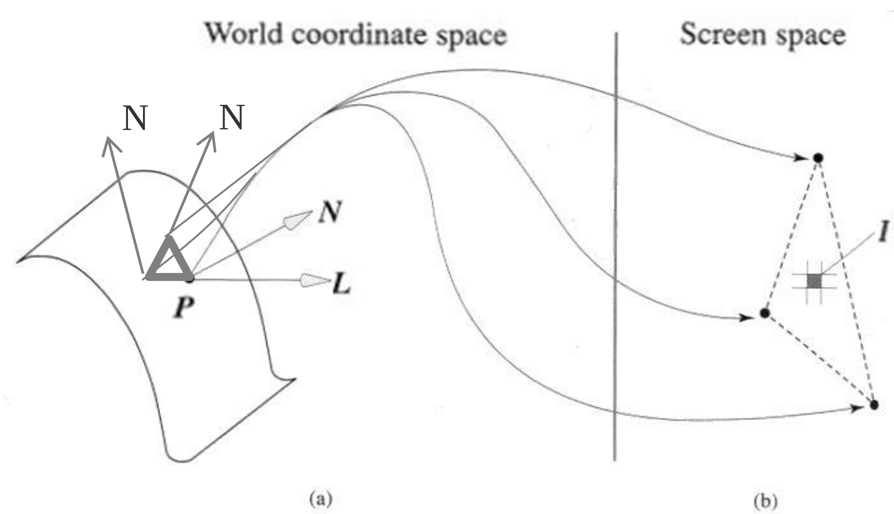


1. Introduction

Previously…

- Local reflection model: Phong: $I_a k_a + I_p(k_d(N.L) + k_s(N.H)^n)$
- Use ambient term to approximate global illumination
- Add on limited global illumination effects: shadows, transparency, reflection (environment) maps





1. Introduction

- Now, we will consider global aspects
- Incorporate indirect reflected and transmitted light
- We'll consider the most used model: *ray tracing*

- Naïve ray tracing only deals with specular to specular light interaction

- Need advanced approaches to model the full 'rendering equation' – see later lecture

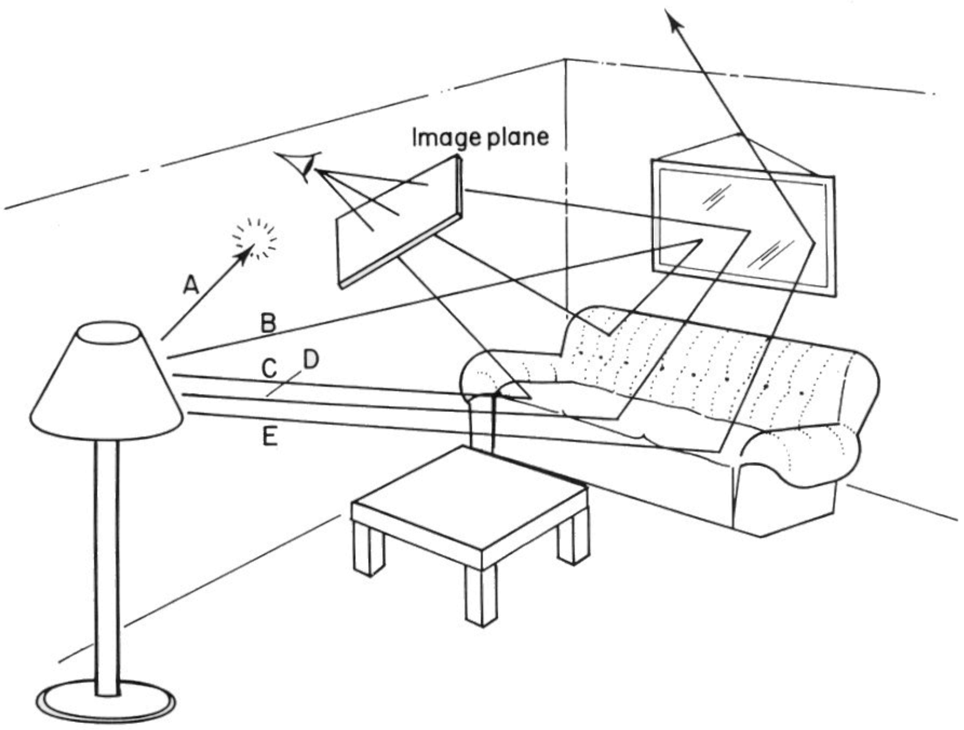


Fig. 5. Some light rays (like A and E) never reach the image plane at all. Others follow simple or complicated routes.

1. Introduction

Rendering pipeline

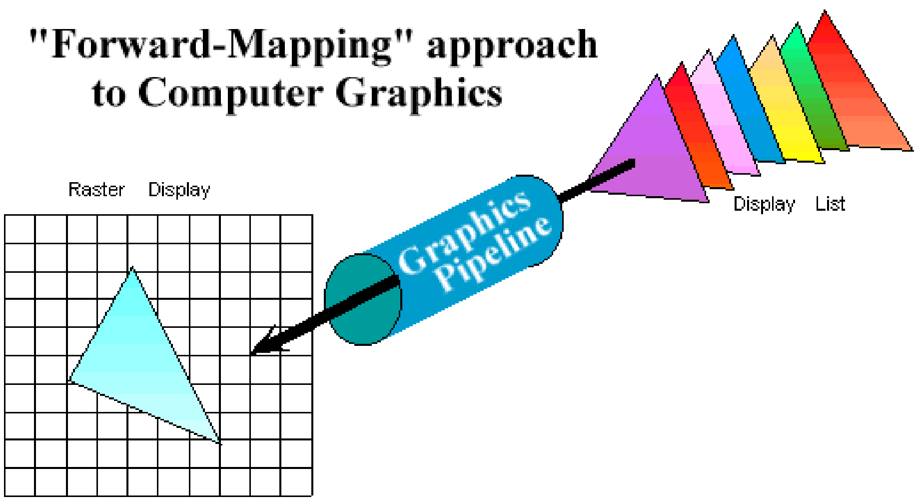
- Project scene to the pixels
- Discretize last
- Algorithm:

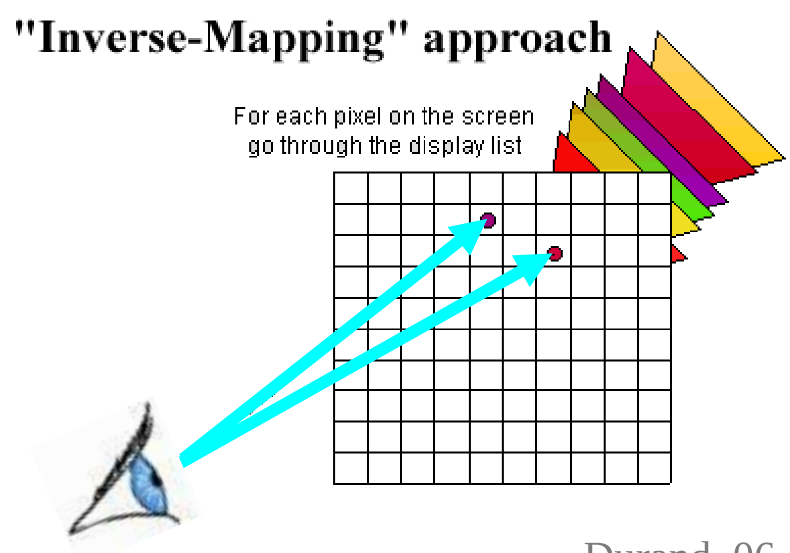
```
For each triangle
  For each pixel
```

Ray casting

- Send pixels to the scene
- Discretize first
- Algorithm:

```
For each pixel
  For each object
```





Durand, 06

1.2 Part 1

Part 1

- Visible surface ray tracing
- Naïve, recursive (Whitted) ray tracing
  - HSR, Shadows, Reflection, Refraction, Recursion

Part 2

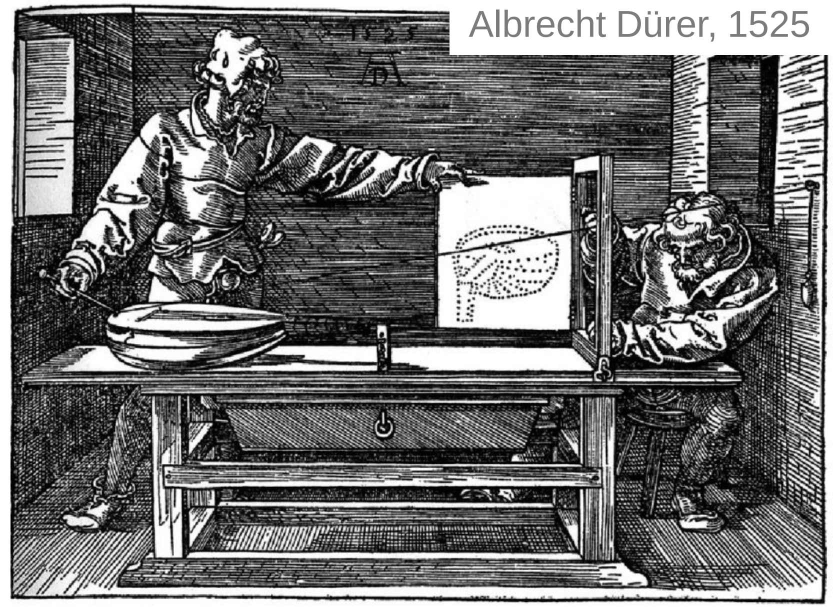
- Intersection calculations
- Speed-up techniques

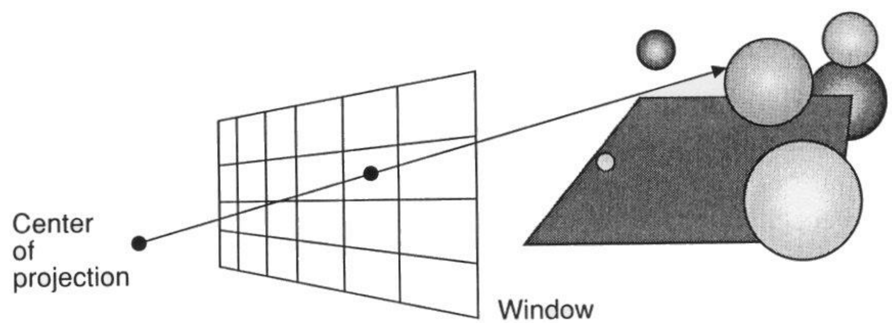
Part 3

- Anti-aliasing
- Advanced techniques

2. Visible surface ray tracing (Appel,68)

- The visibility of surfaces is determined by tracing imaginary rays of light from the viewer’s eye to the objects in the scene
- Three parts:
    - Initial ray direction – different for each pixel
    - Hidden surface removal – which surface is closest?
    - Shadows – can a surface ‘see’ the light source

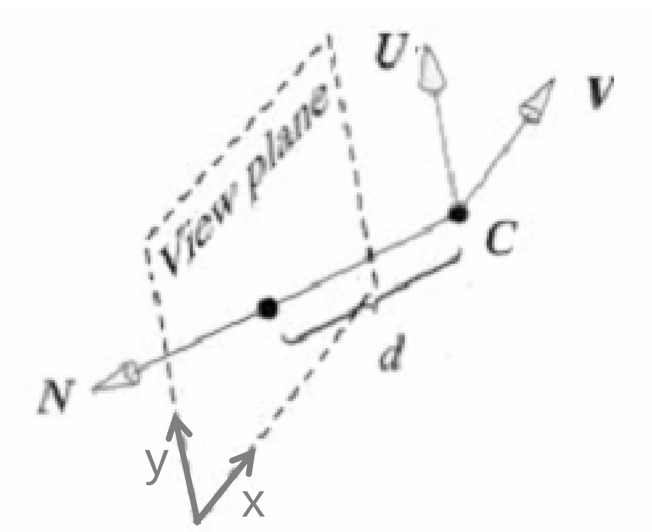

Albrecht Dürer, 1525

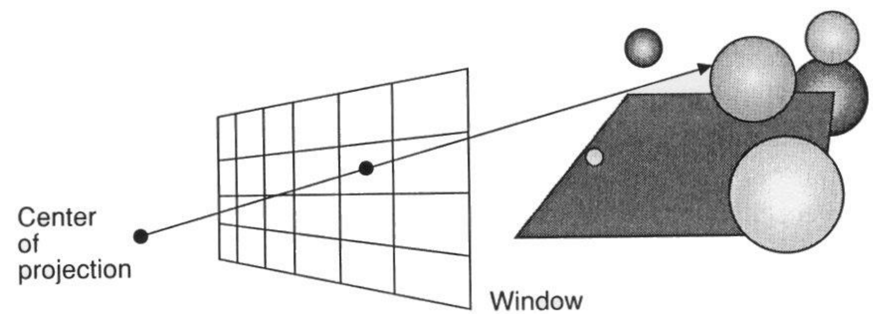


Appel, A. “Some Techniques for Shading Machine Renderings of Solids.” *Proceedings of the Spring Joint Computer Conference*. 1968, pp. 37-45.

2.1 Initial ray direction

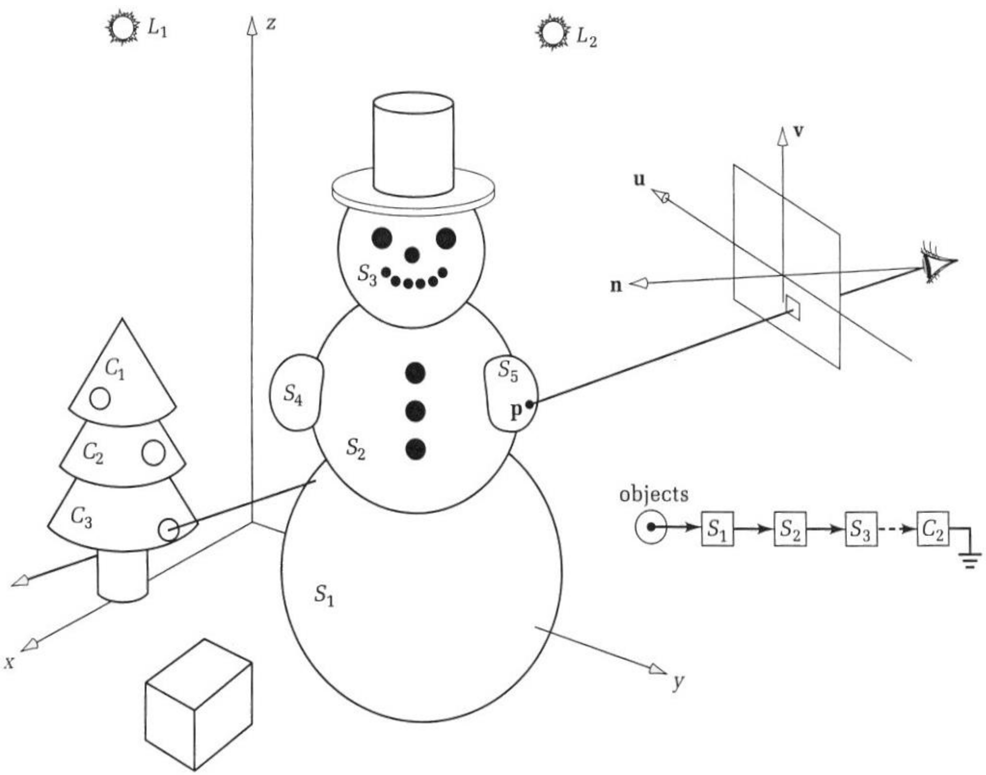
- Position of camera and view plane are defined
  - Viewplane is at distance `d` from camera along `N`
- A ray is traced from the camera through **each** screen pixel (=wxh rays)
  - Ray start position and direction calculated using `C`, `d`, `N`, `x`, `V`, `y` and `U`



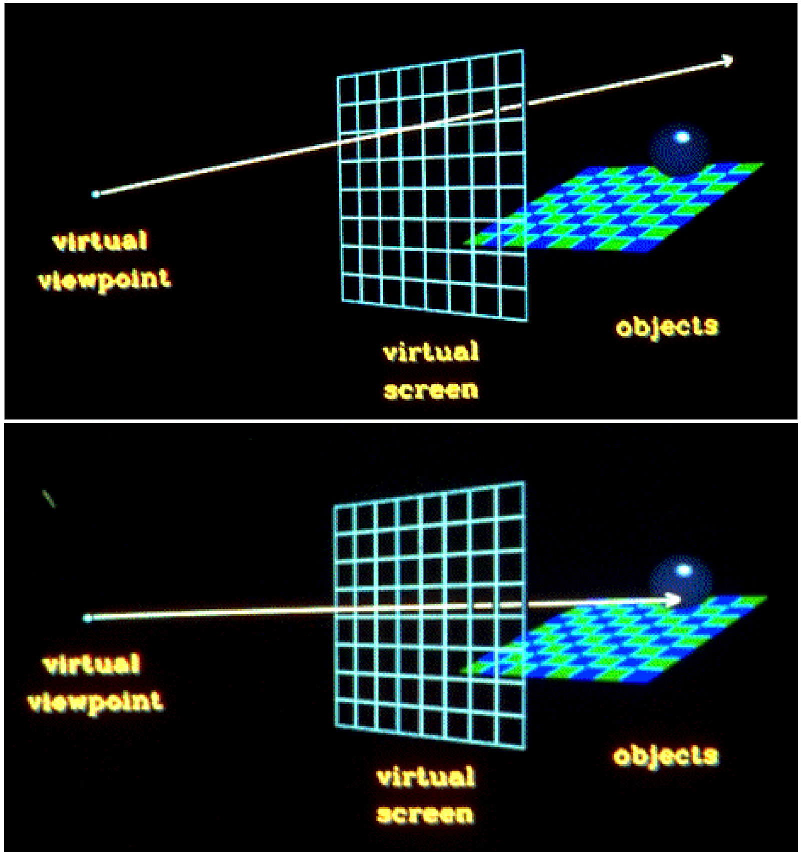


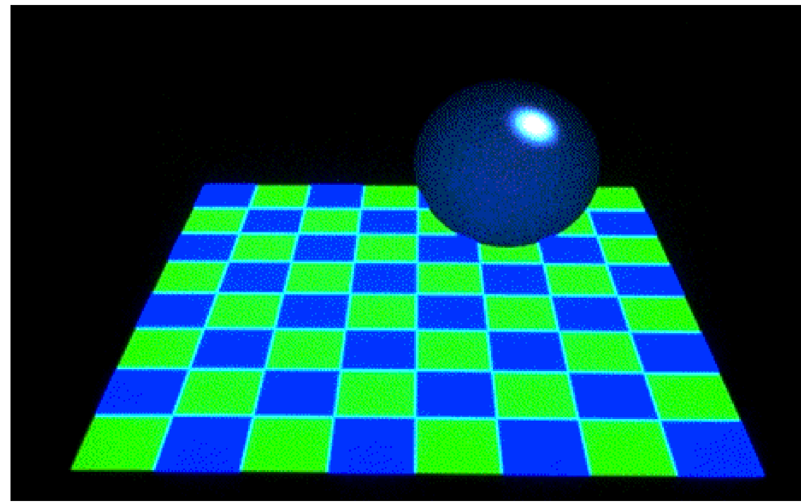
2.2 Hidden surface removal

- Each ray must be tested against all objects in the scene
  - (wxh rays) x (m objects)
- List of intersected objects is found
- Closest intersection is chosen
- For p pixels and m objects, that is p x m ray-object intersection calculations
  - If p=1000x1000 and m=100, then 100,000,000 ray-object intersection calculations
- If n polygons per object, that is p x m x n ray-polygon intersection calculations
  - If p=1000x1000 and m=100 and n=1000, then 100,000,000,000 ray-polygon intersection calculations



2.2 HSR

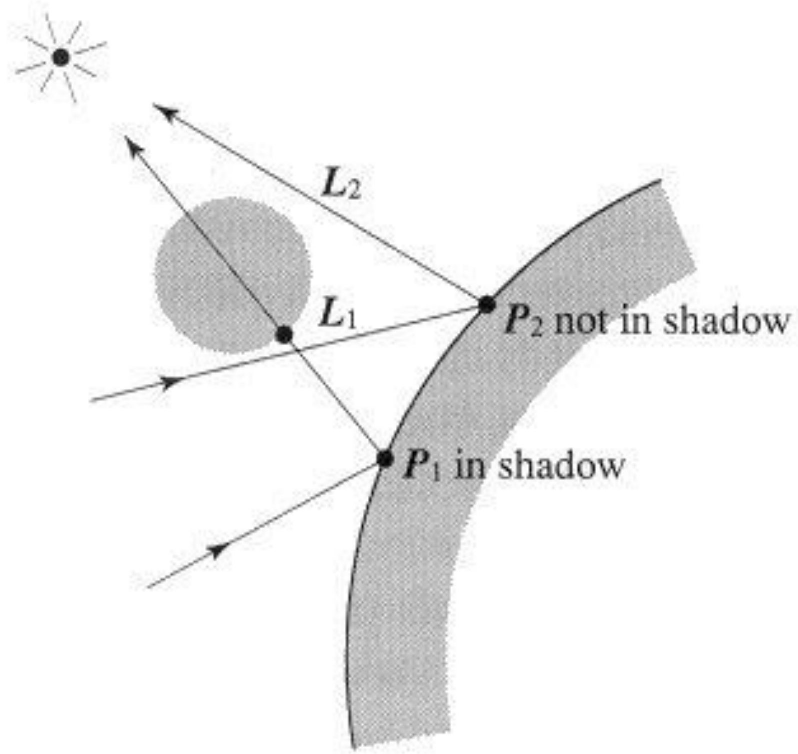




- Apply a lighting equation at the intersection point, e.g. Phong

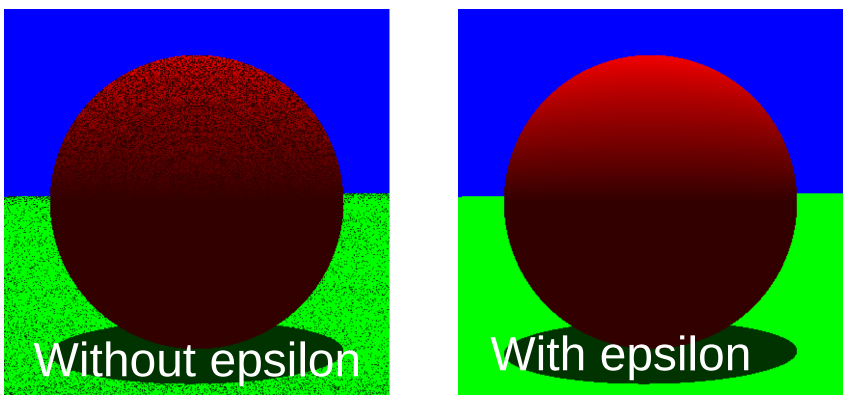
2.3 Shadows

- At a surface intersection P, a shadow feeler ray (or light ray) is spawned towards the light source
  - (for each light source)
- **All** objects in the scene are tested against this ray
  - Intersection calculations
- If a shadow feeler intersects an object on the way to the light source, then the point P is in shadow
  - ‘Early out’ to stop intersection tests
- Only hard-edged shadows are produced by this simple approach

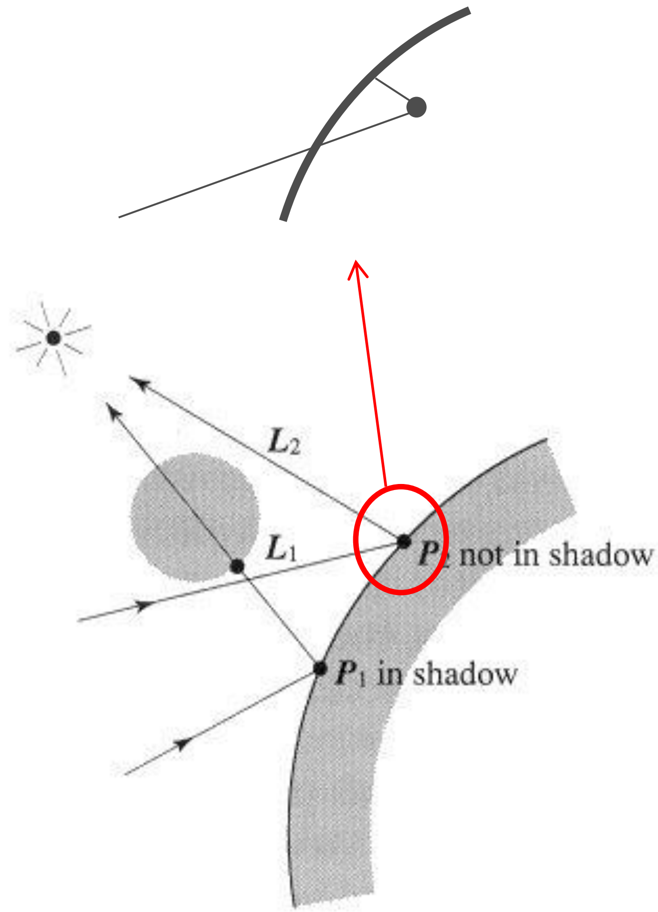


2.3 Shadows

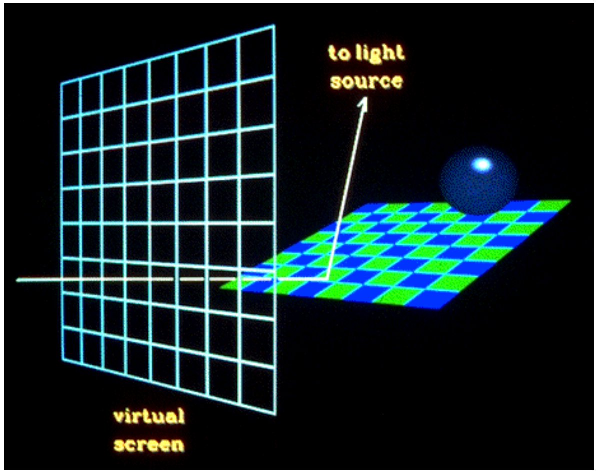
- Beware of numerical accuracy
- Intersection point is inside the shape because of numerical inaccuracy, therefore light ray will intersect same object, resulting in a ‘shadow’ (speckled appearance)
- *Solution*: add a small amount to make sure intersection point is outside object

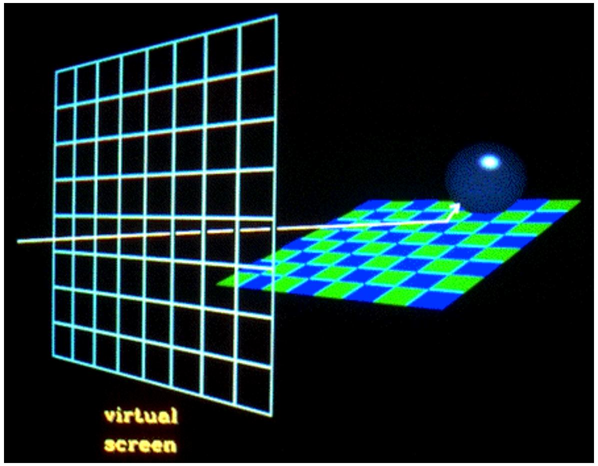


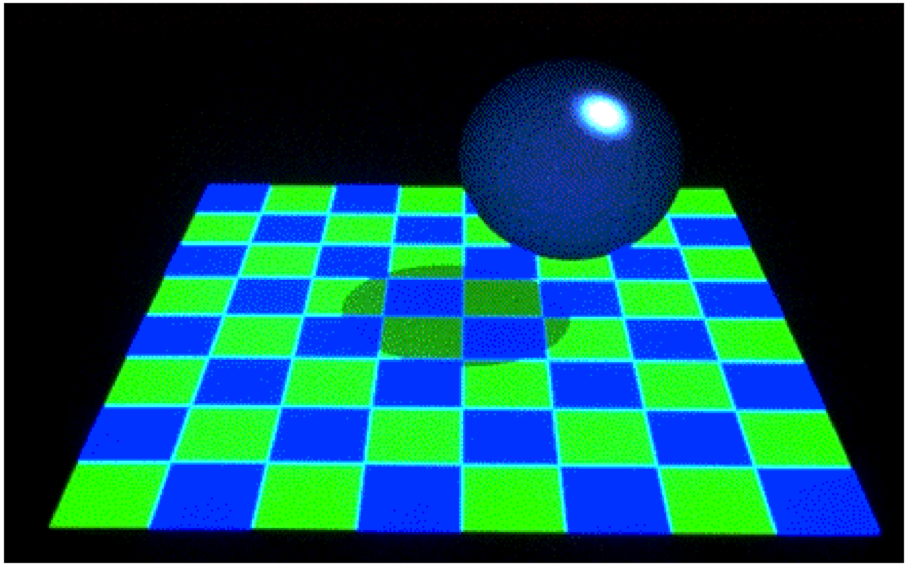
Durand, 06



2.3 Shadows







- If in shadow, alter light calculation
  - (guess an amount to reduce the light)

2.4 Results (Appel,68)

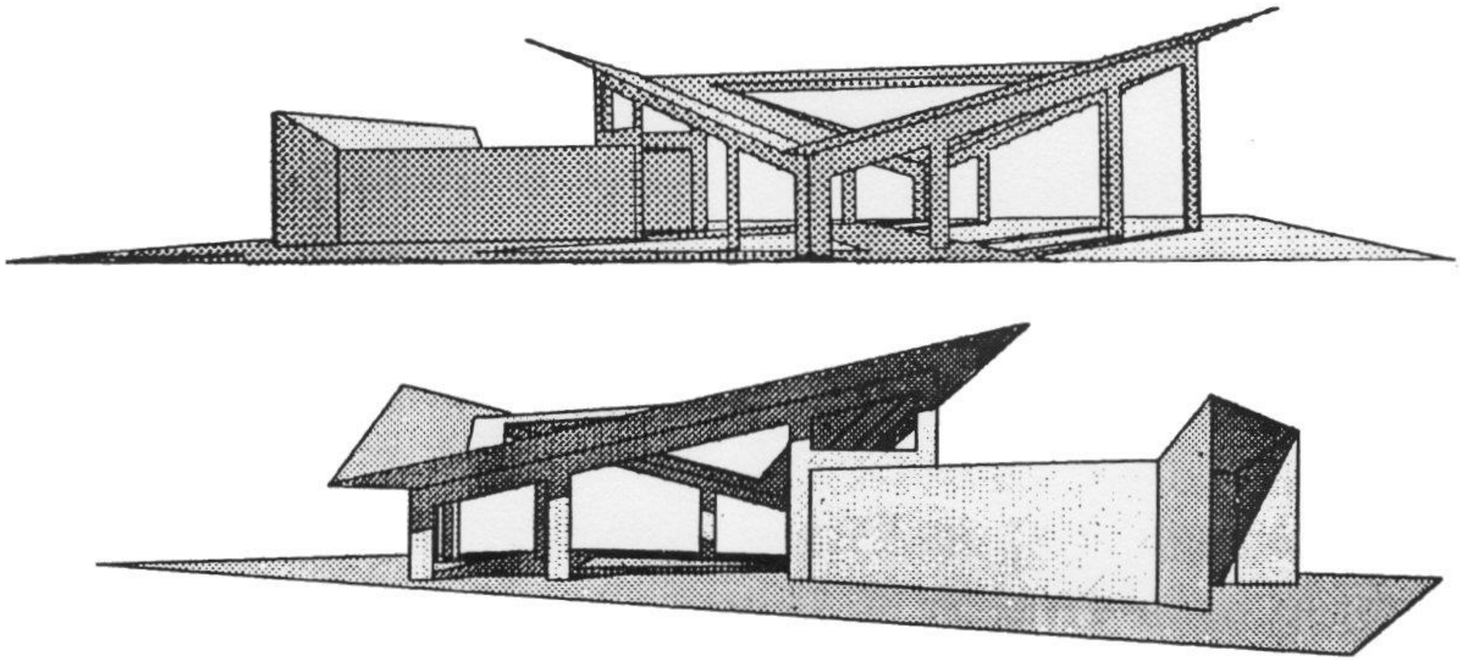


Fig. 16.52 Early pictures rendered with ray tracing. (Courtesy of Arthur Appel, IBM T.J. Watson Research Center.)

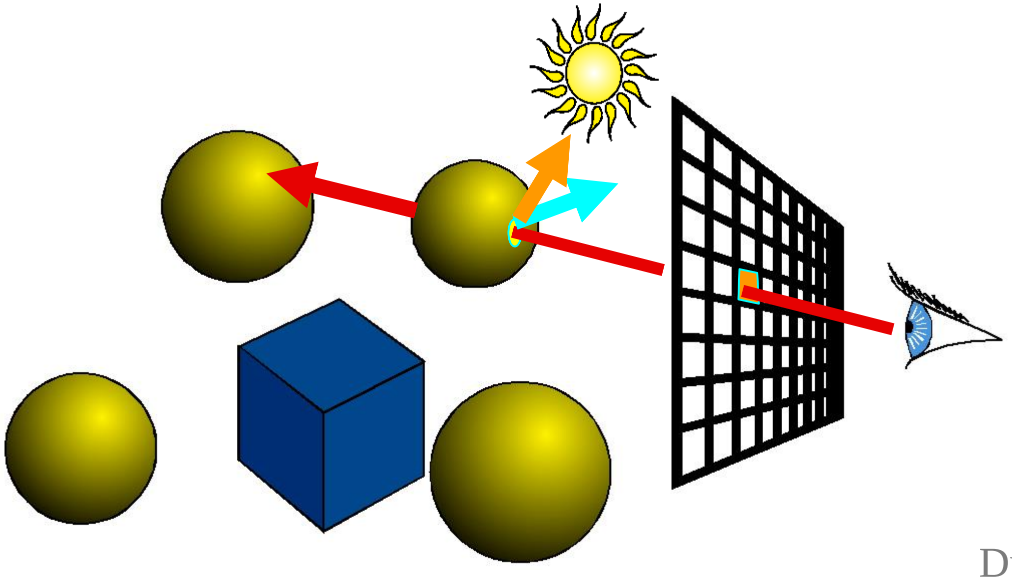
Appel, A. "Some Techniques for Shading Machine Renderings of Solids." *Proceedings of the Spring Joint Computer Conference*. 1968, pp. 37-45.

2.5 As an algorithm…

```
For every pixel
   Construct a ray from the eye
   // find closest intersection point
   For every object in the scene
        Find intersection with the ray
        Keep if closest
   // then shade using the closest intersection point
   Shade depending on light and normal vector
```

Key parts:

- Intersection testing
- Calculating normal for use in lighting calculation



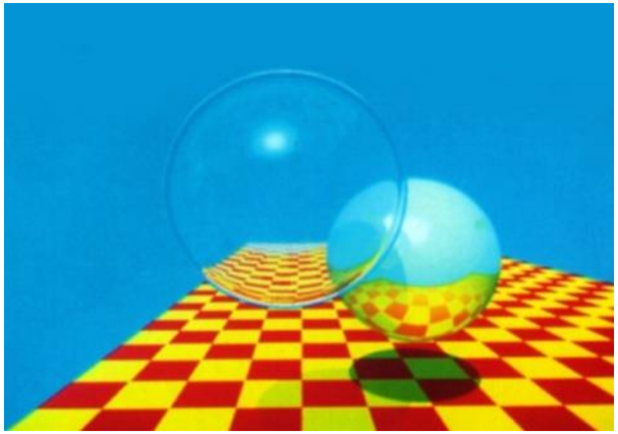
Durand, 06

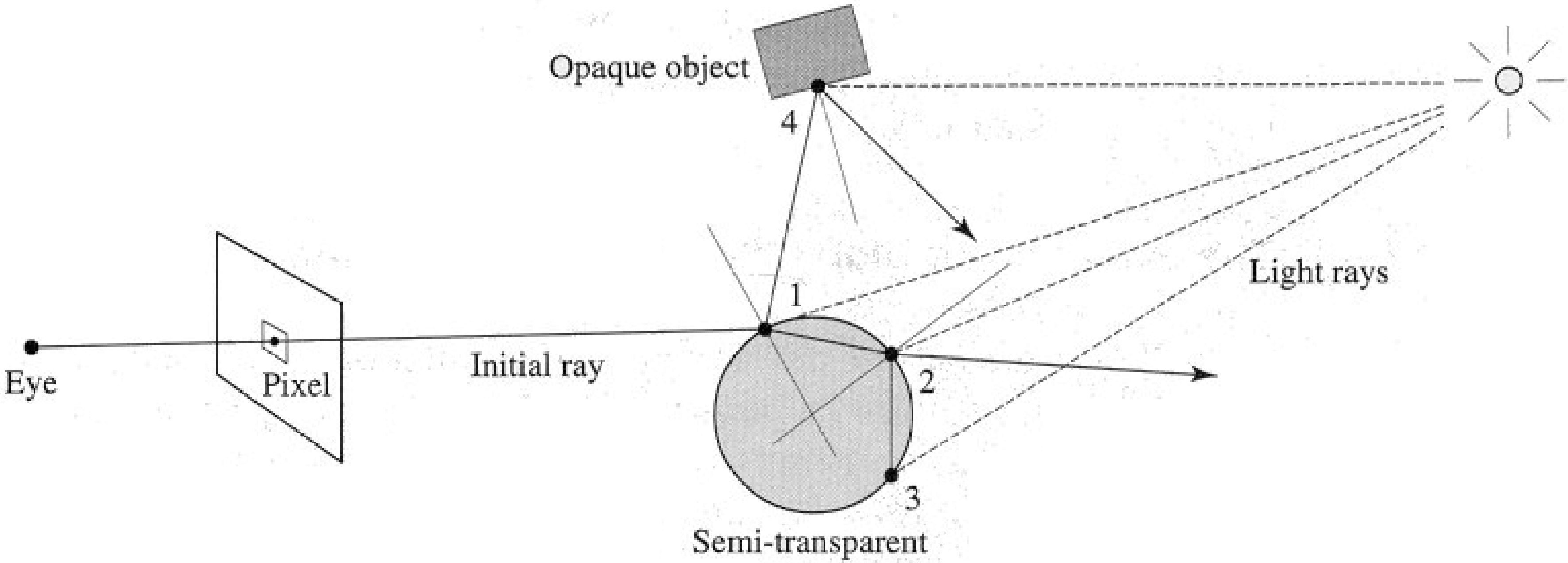
Turner Whitted: An Improved Illumination Model for Shaded Display. Commun. ACM 23(6): 343-349 (1980)

3. Recursive ray tracing (Whitted, 80)

- Referred to as **standard naïve ray tracing**
- An elegant combination of:
  - Hidden surface removal;
  - Shading due to direct illumination;
  - Global specular interaction effects – reflected rays and refracted rays;
  - Shadow (geometry) computation.

Whitted, 1980





3.1 Stages

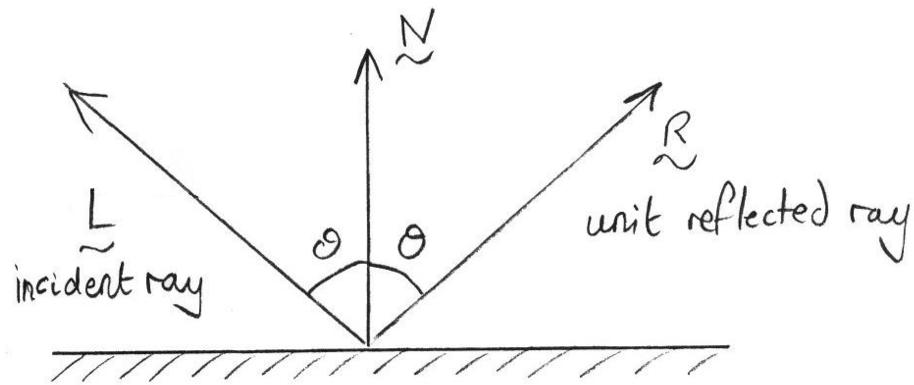
- Initial ray direction – as before
- Hidden surface removal – as before
- Reflected and refracted rays
- Lighting model
- Shadows – as before
- Recursion

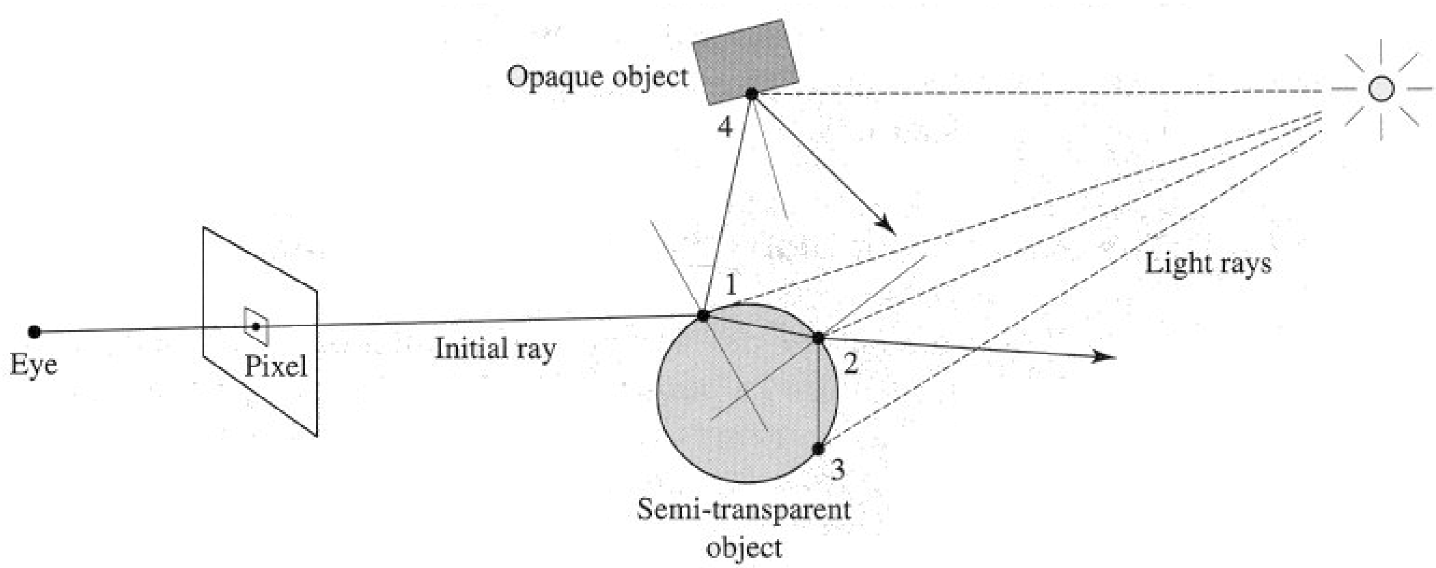
3.2 Reflected ray

- At an intersection point, a reflected ray is formed

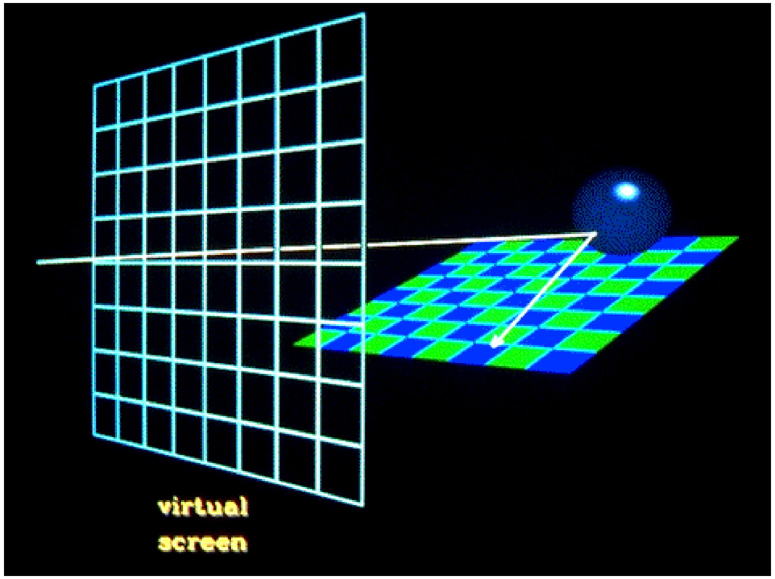
  R = 2(**N**.**L**)**N** – **L**

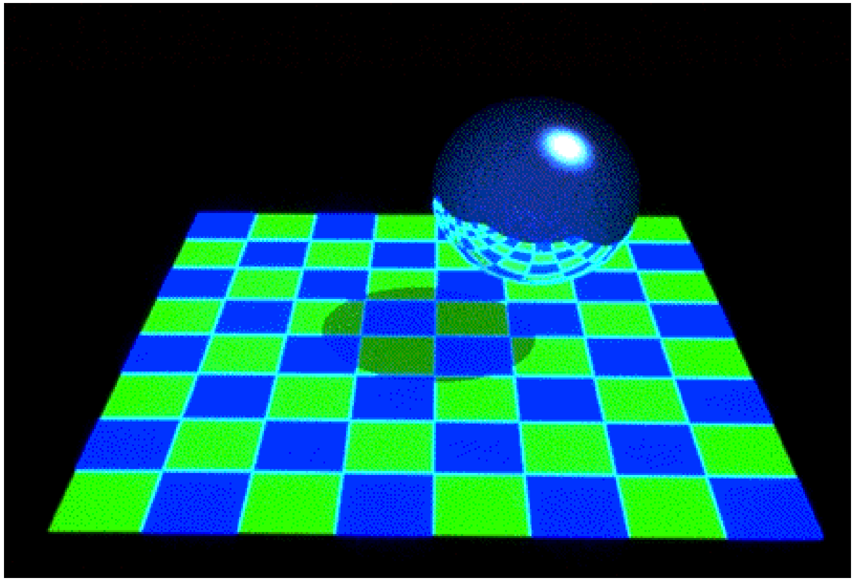
- This can then be traced further… recursion





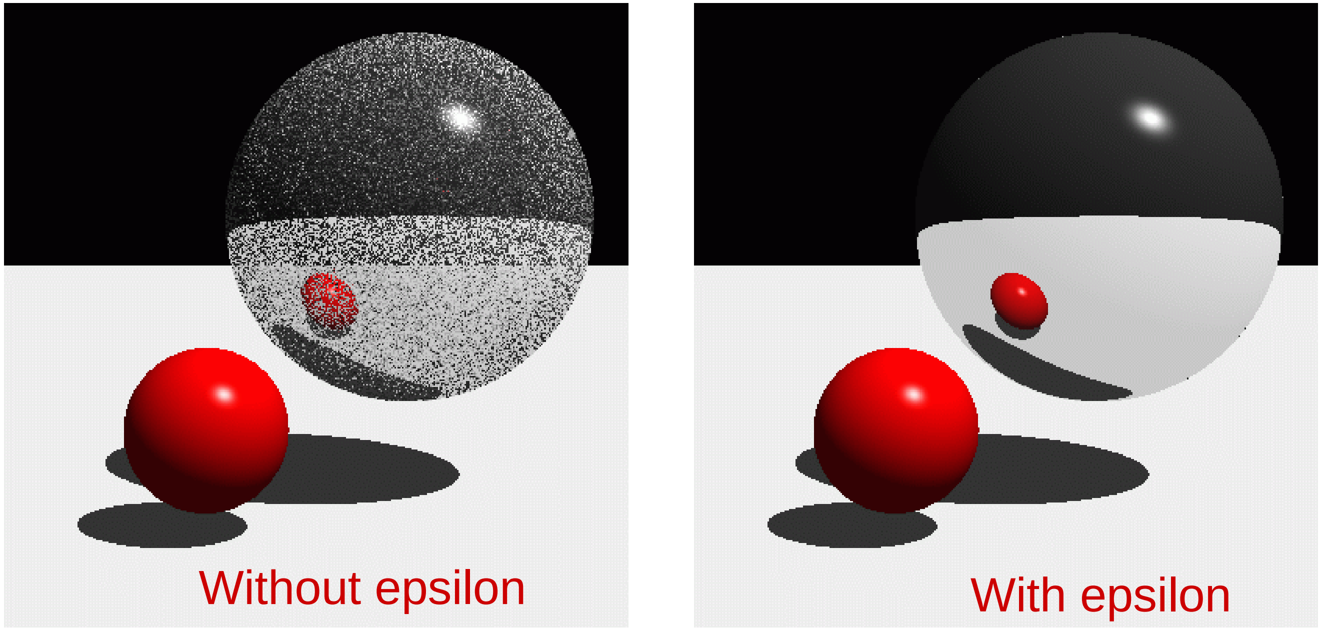
3.2 Reflected ray





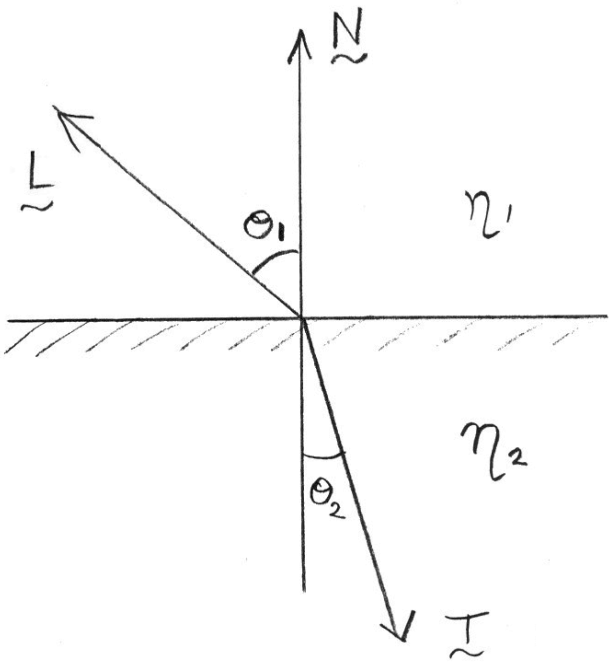
3.2 Reflected ray

- Again, beware numerical accuracy



Durand, 06

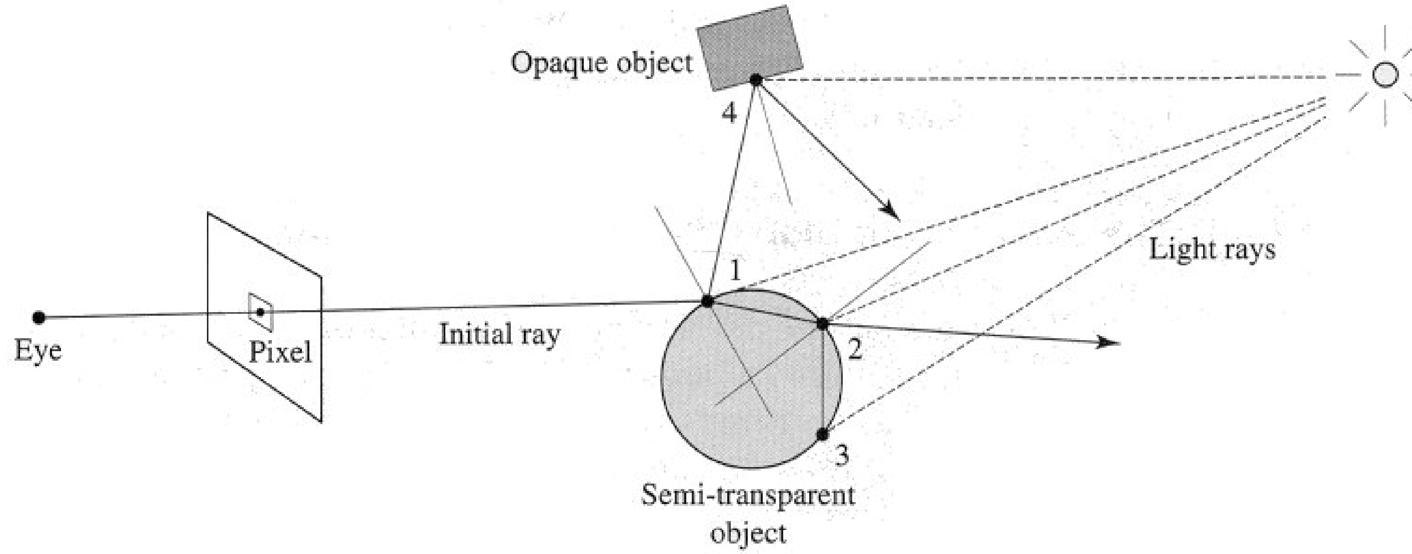
3.3 Refracted ray



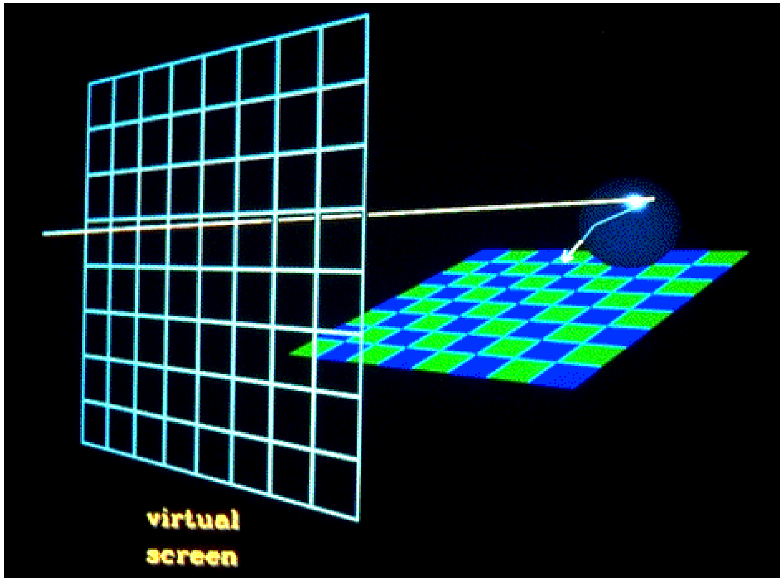
- Depending on the nature of the object, a refracted ray may be required.
- Snell's law $\quad \eta_1 \sin\theta_1 = \eta_2 \sin\theta_2$

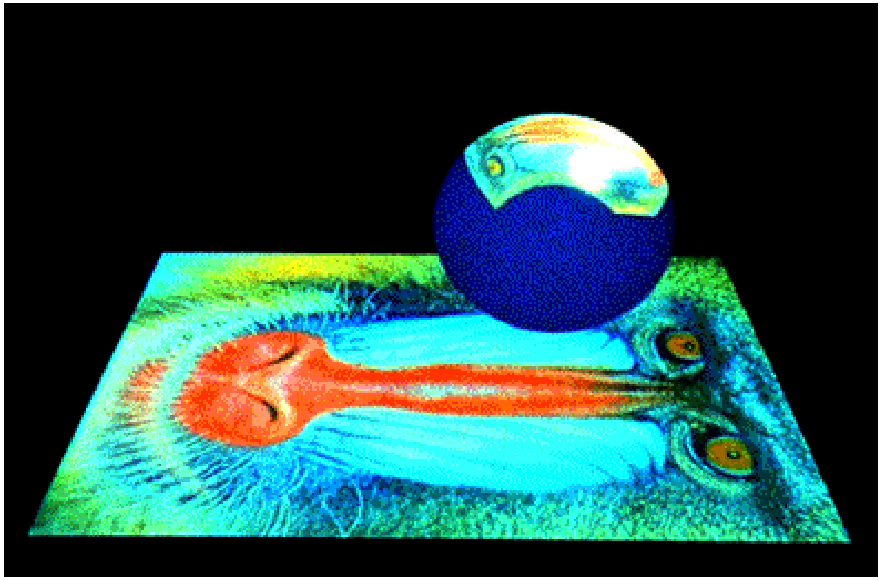
$$\mathbf{T} = -\frac{\eta_1}{\eta_2}\mathbf{L} - \left(\cos\theta_2 - \frac{\eta_1}{\eta_2}\cos\theta_1\right)\mathbf{N}$$

- This is traced further… recursion



3.3 Refracted ray





3.3 Refracted ray

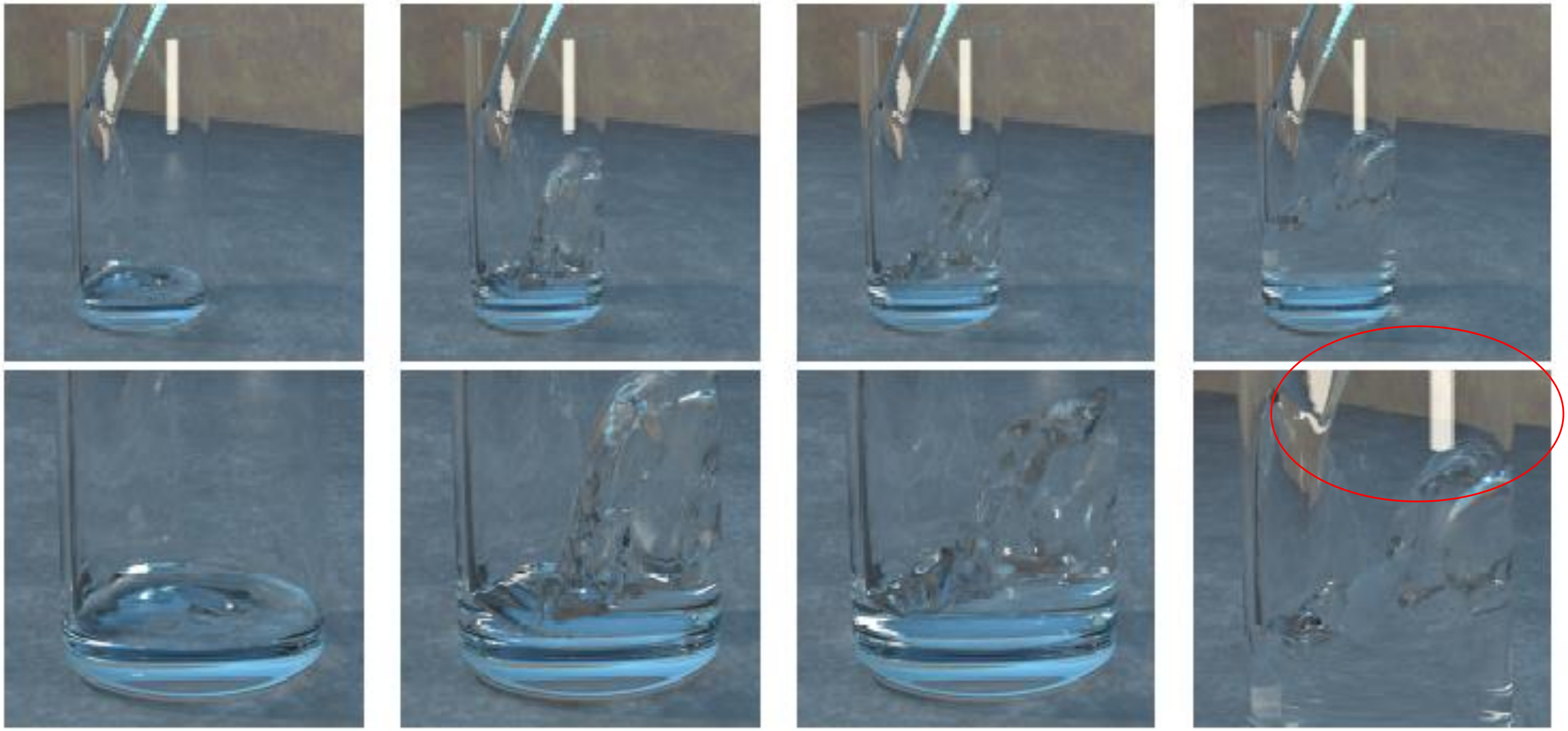


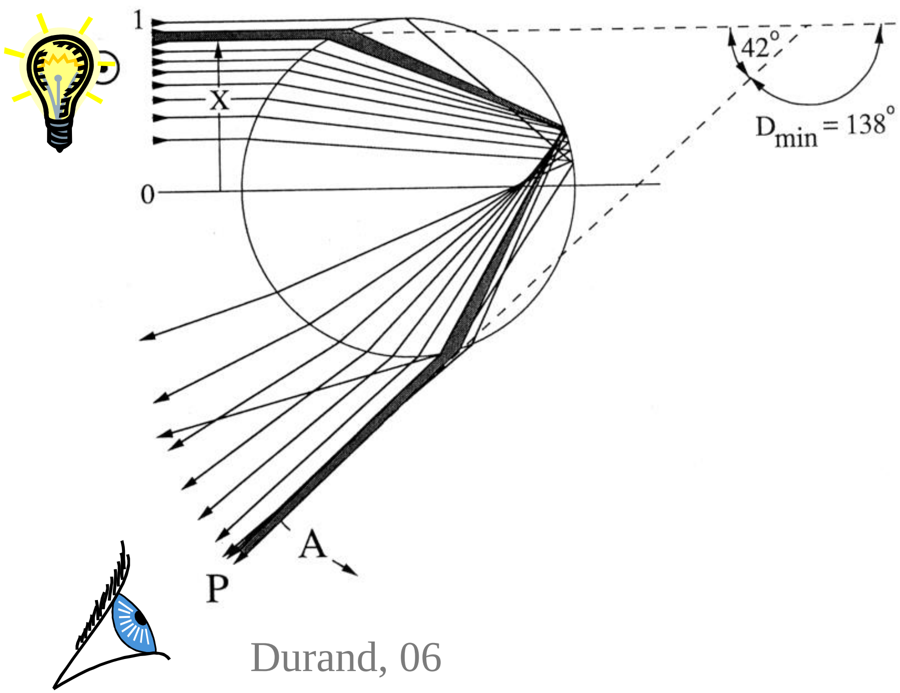
Figure 7: **Water being poured into a clear, cylindrical glass (55x55x120 grid cells). Our method makes possible the fine detail seen in the turbulent mixing of the water and air.**

Demo 1, Demo 2

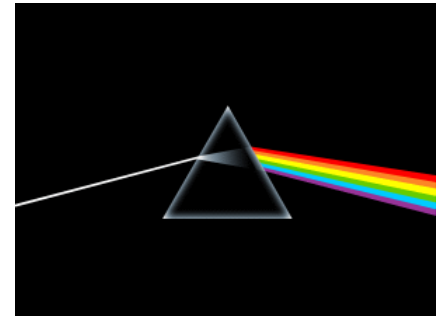
Douglas Enright, Steve Marschner, and Ronald Fedkiw , "Animation and Rendering of Complex Water Surfaces", ACM Transactions on Graphics (Proc. SIGGRAPH 2002)

Aside: Refraction depends on wavelength

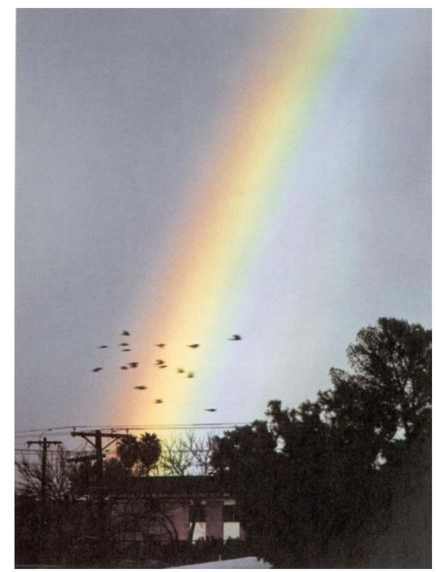
- Refraction depends on wavelength → prism
- Rainbow is caused by
  refraction + internal reflection + refraction
- Maximum for angle approx 42 degrees



Durand, 06



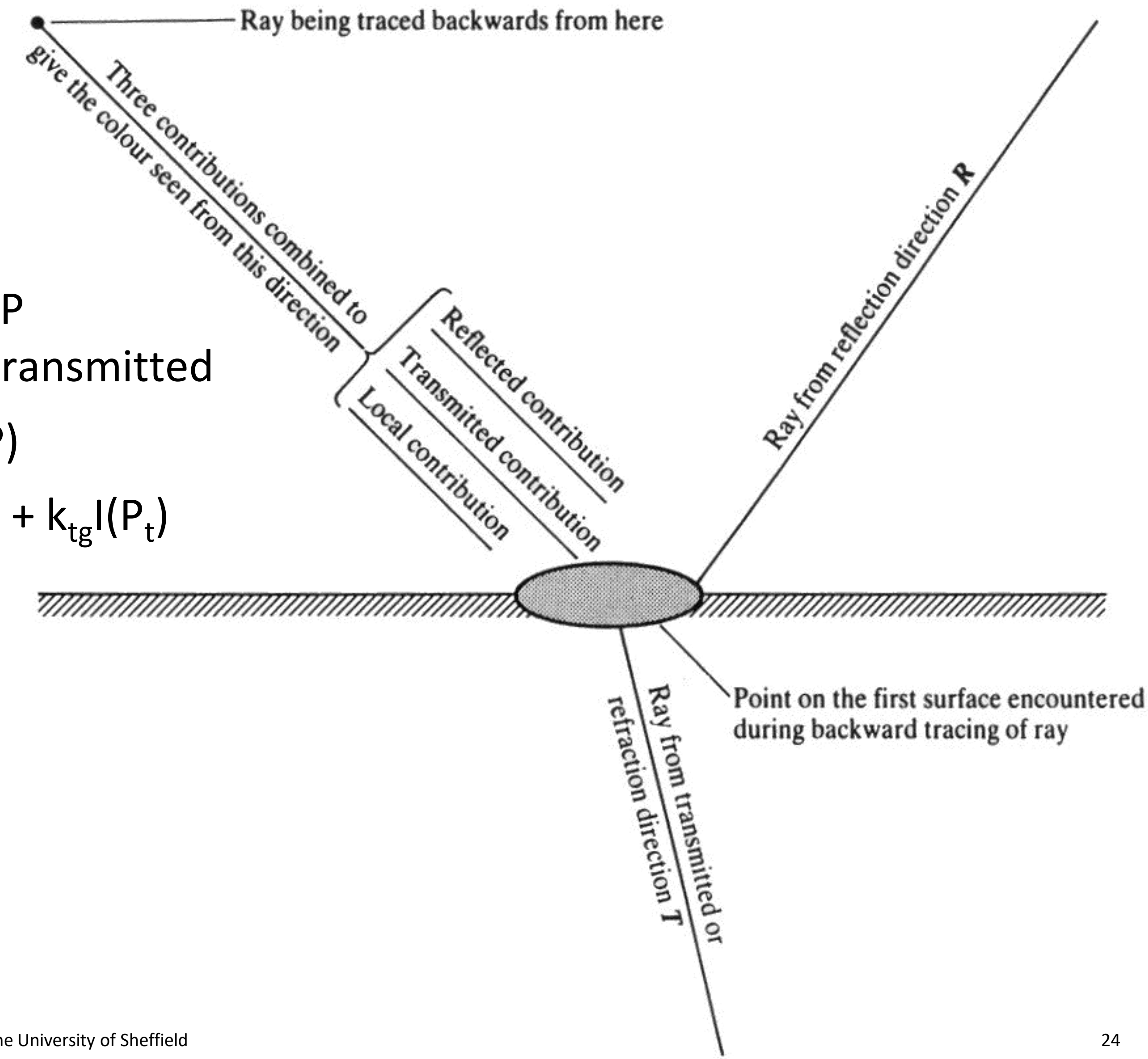
Pink Floyd, *The Dark Side of the Moon*, 1973



From "Color and Light in Nature" by Lynch and Livingstone

3.4 Combining the lighting model components

- Intensity at hit point P
  = local + reflected + transmitted
- $I(P) = I_{local}(P) + I_{global}(P)$
- $I(P) = I_{local}(P) + k_{rg}I(P_r) + k_{tg}I(P_t)$

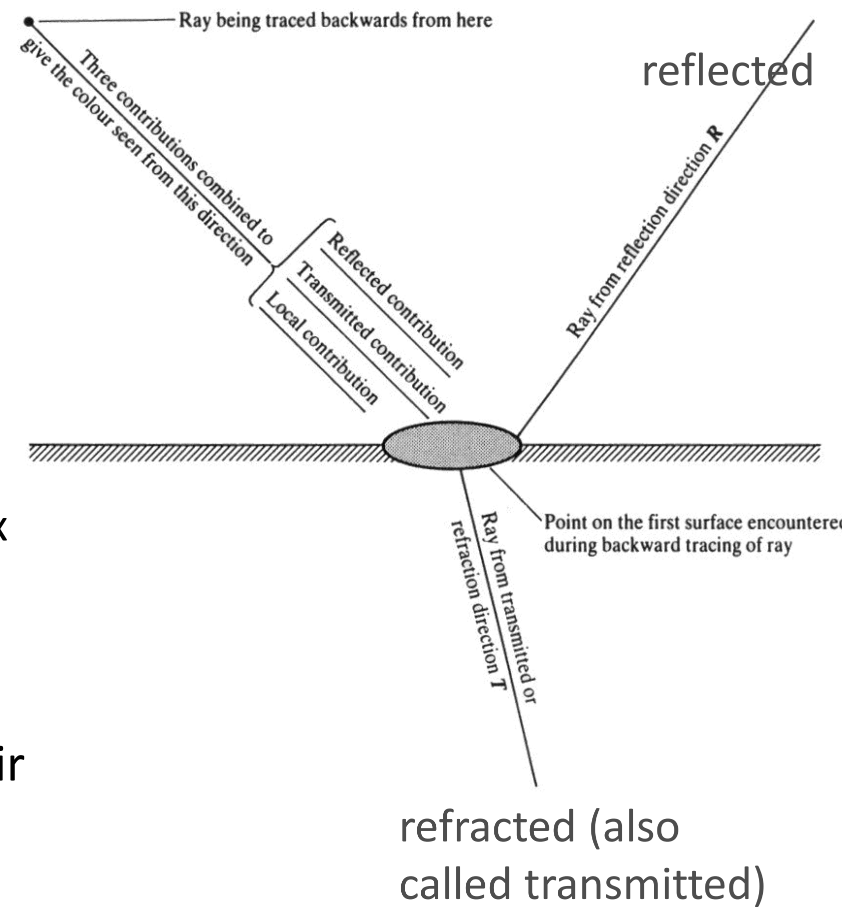


3.4 Combining the lighting model components

- Local model – Phong:
  - I_{local} = ambient + diffuse + specular
    $= I_a k_a + I_p(k_d(N.L) + k_s(N.H)^n)$

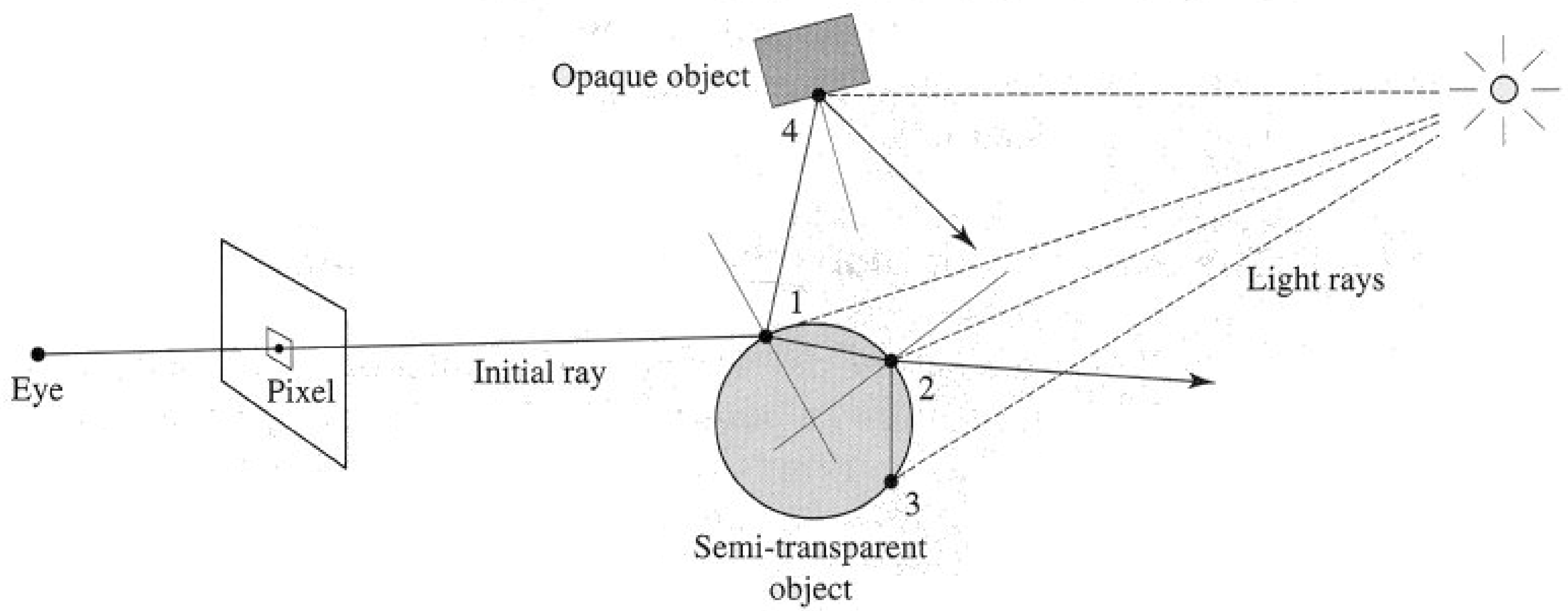
    (or, if light source 'behind' a transparent object
    $= I_a k_a + I_p(k_d(N.L) + k_t(N.H')^m)$
    where H' involves the refractive index of the material)

- In this recursive process, $I_{reflected}$ and $I_{transmitted}$ invoke similar terms at their next respective surface intersections



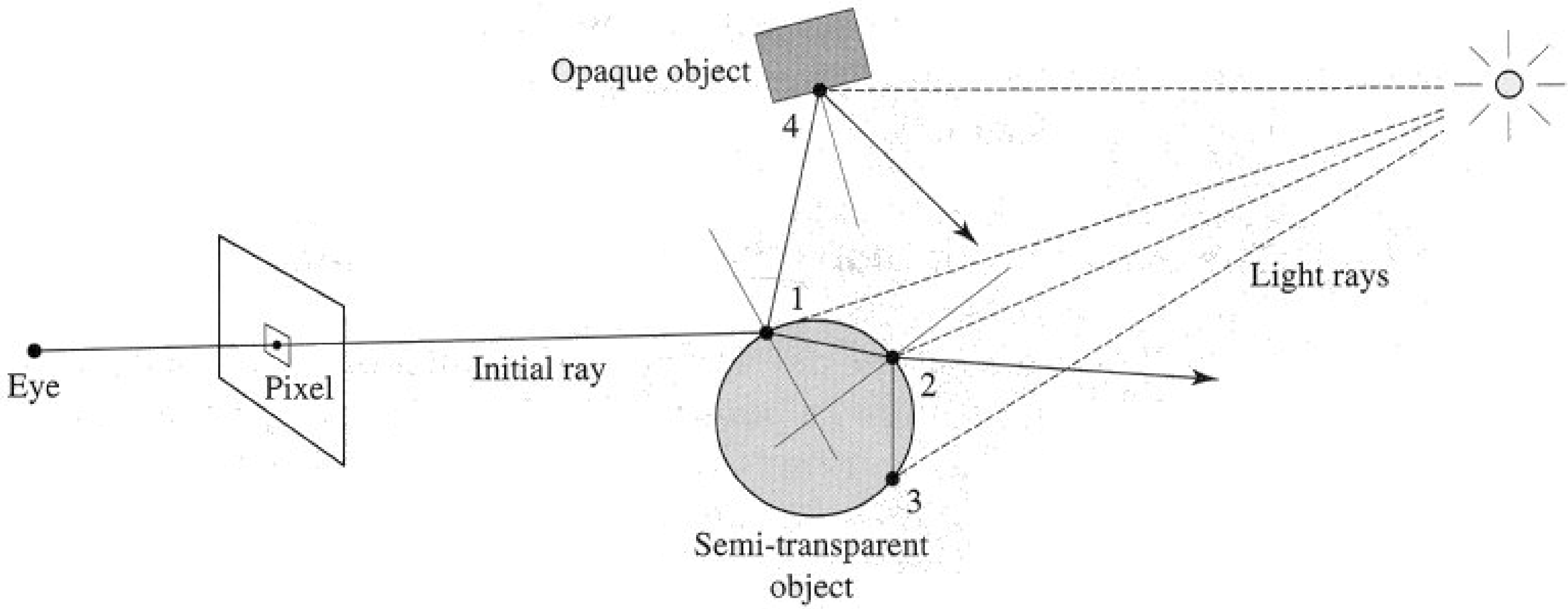
3.4 Combining the lighting model components

- At intersection point 1 (and similar at other intersection points):
  - Local calculation using Phong
  - Reflected component from 4
  - Refracted component from 2
  - Also, light ray (shadow feeler) component



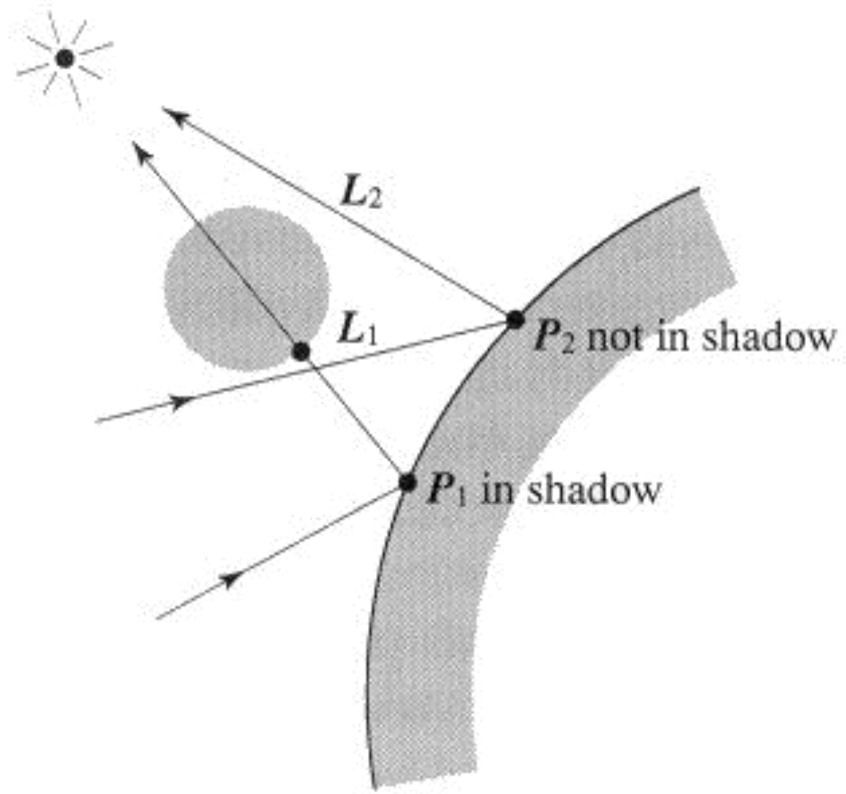
3.4 Combining the lighting model components

- Because of the mix of local and global components, we have a blurred specular highlight and a perfect specular reflection
- The diffuse term is local
  - If we hit a diffuse object, then the recursion can terminate.
- (We could model diffuse-diffuse effects using Monte Carlo methods – see later lecture.)



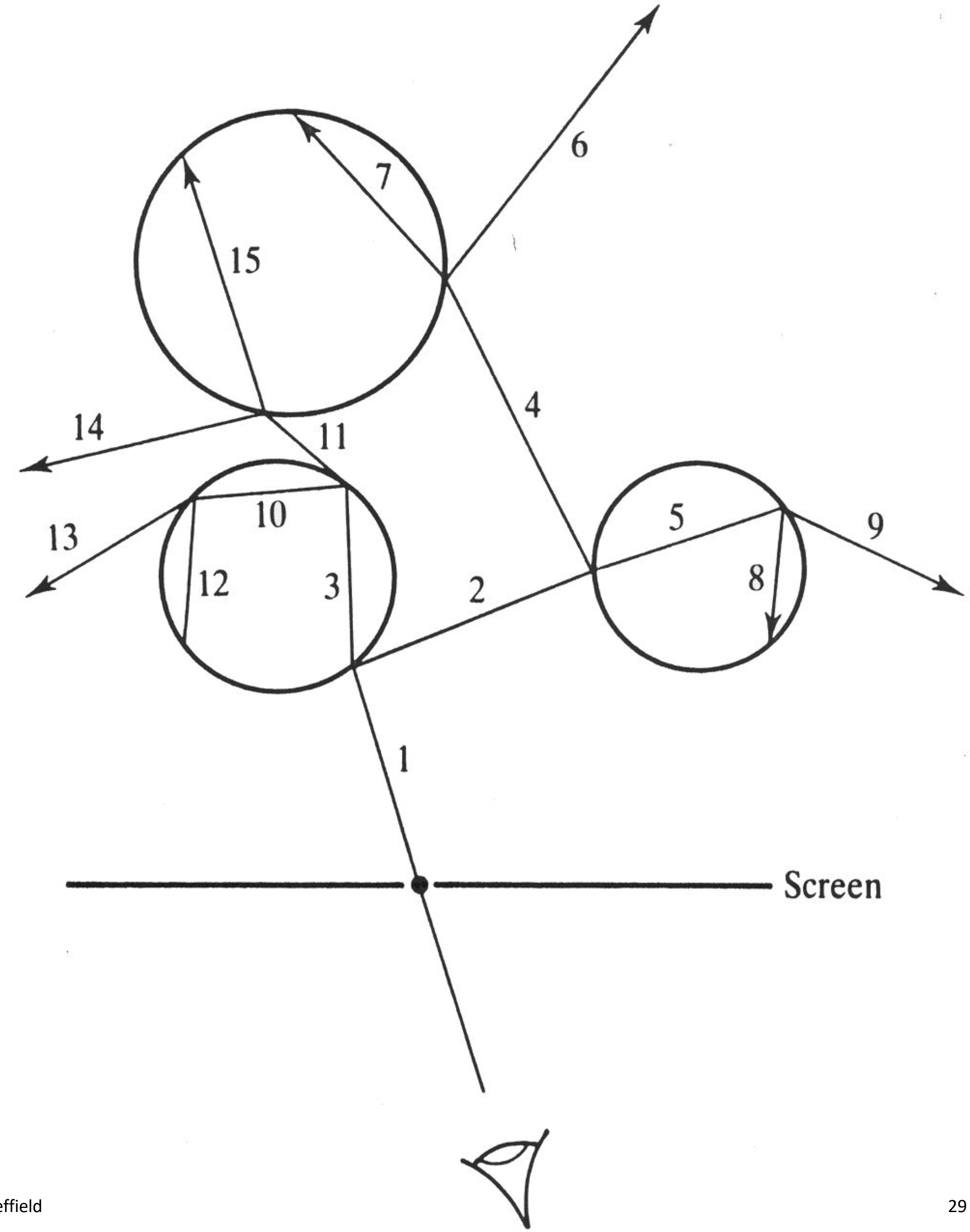
3.5 Shadows

- At a surface intersection a shadow feeler ray (or light ray) can be spawned towards the light source
- If a shadow feeler intersects an opaque object on the way to the light source, I_{local} is set to the ambient term for that light source (or diffuse term is modulated)
  - For a transparent object I_{local} is attenuated
- Only hard-edged shadows are produced by this simple approach
  - see distributed ray tracing for penumbra



3.6 Recursion

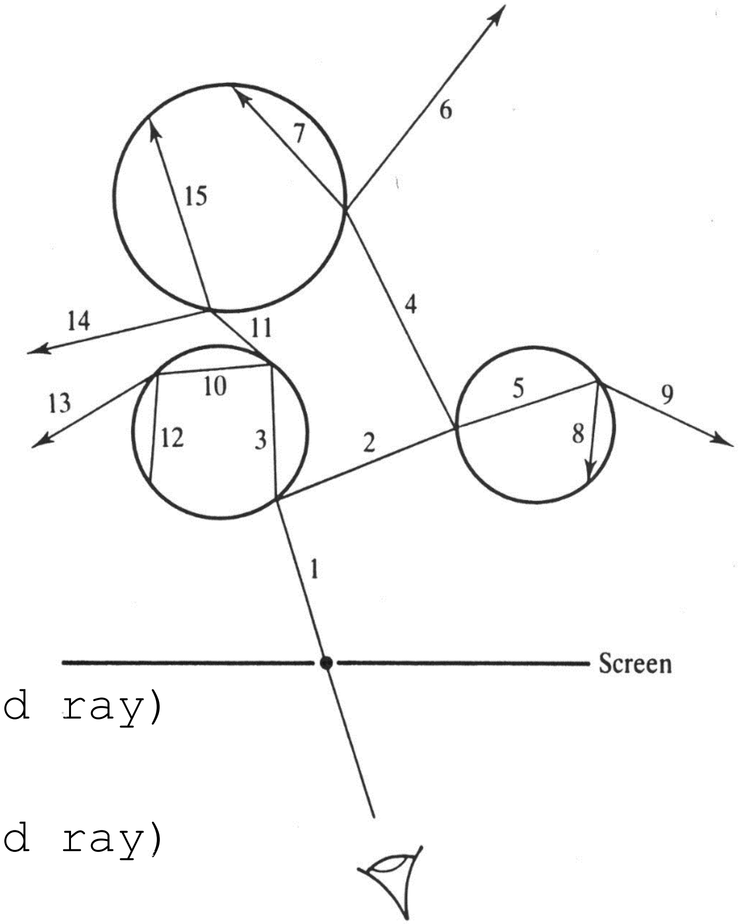
- For a single ray we can follow the reflections and refractions
- This is a recursive process since following a ray may give rise to following further rays, and so on...

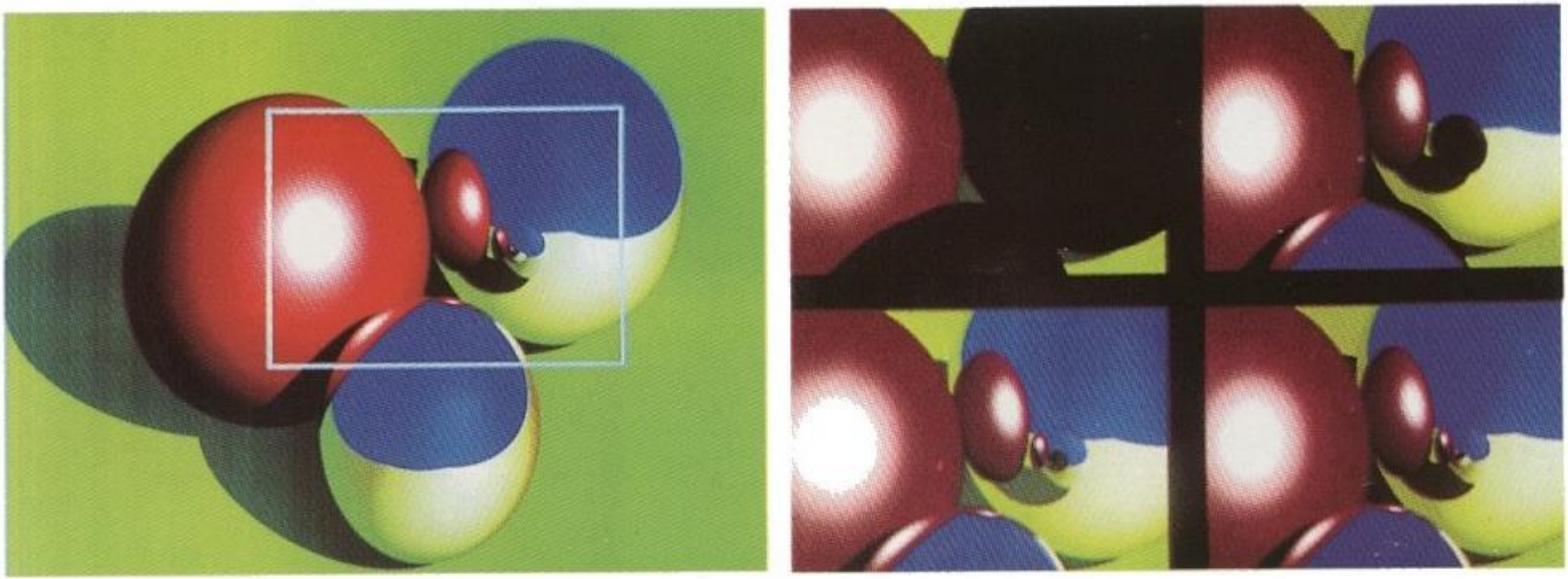


3.6 Recursion

shootRay (ray structure)

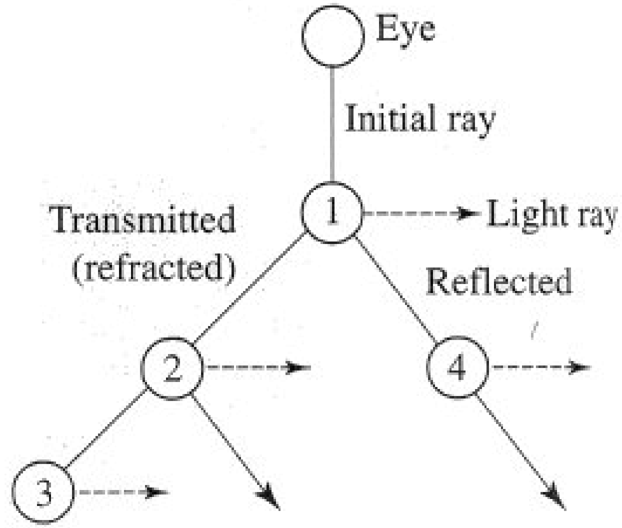
```
intersection test for all objects;
if  ray intersects objects {
  get closest object intersection;
  for every light, cast shadow ray;
  get normal at intersection point;
  calculate local intensity (Ilocal);
  if (reflection)
    calculate and shootRay(reflected ray)
  if (refraction)
    calculate and shootRay(refracted ray)
  Intensity at hit point P = local
    + reflected + transmitted
}
```

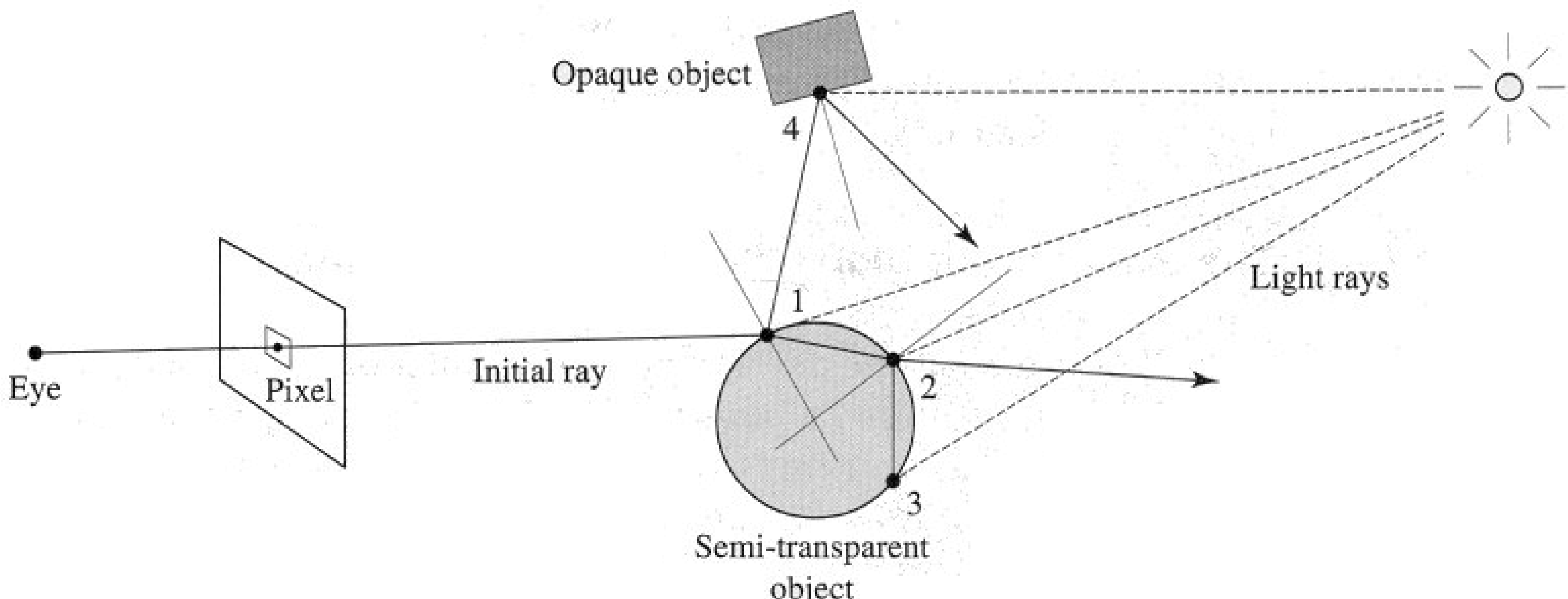




32 Ray tracing spheres.
(*bottom left*) Ray tracing to a depth of 6. (*bottom right*) Part of the above image traced to a depth of 1, 2, 3 and 4.

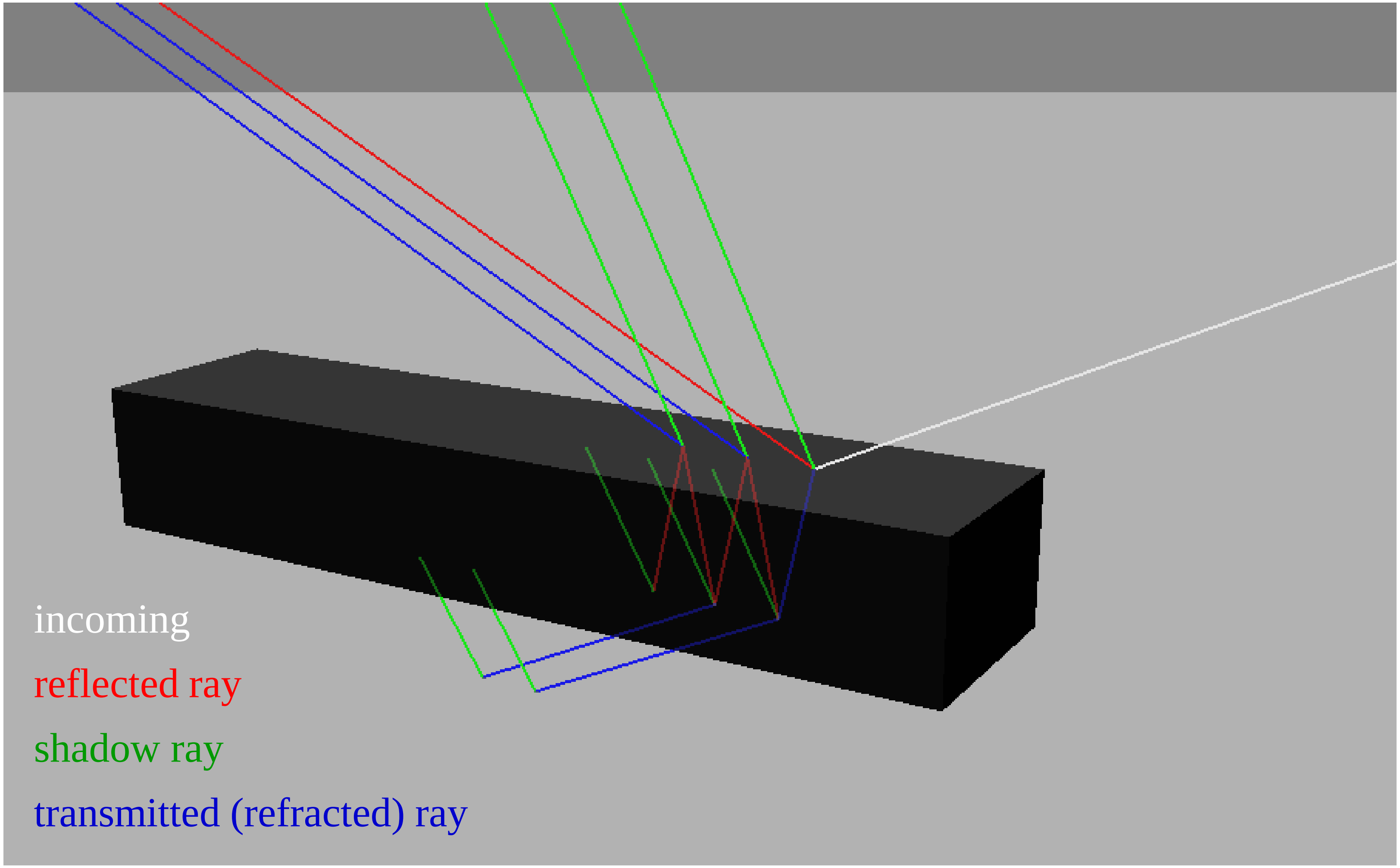
3.6.1 The ray tree





3.6.1 The ray tree

- Visualize the ray tree for single pixel in an image

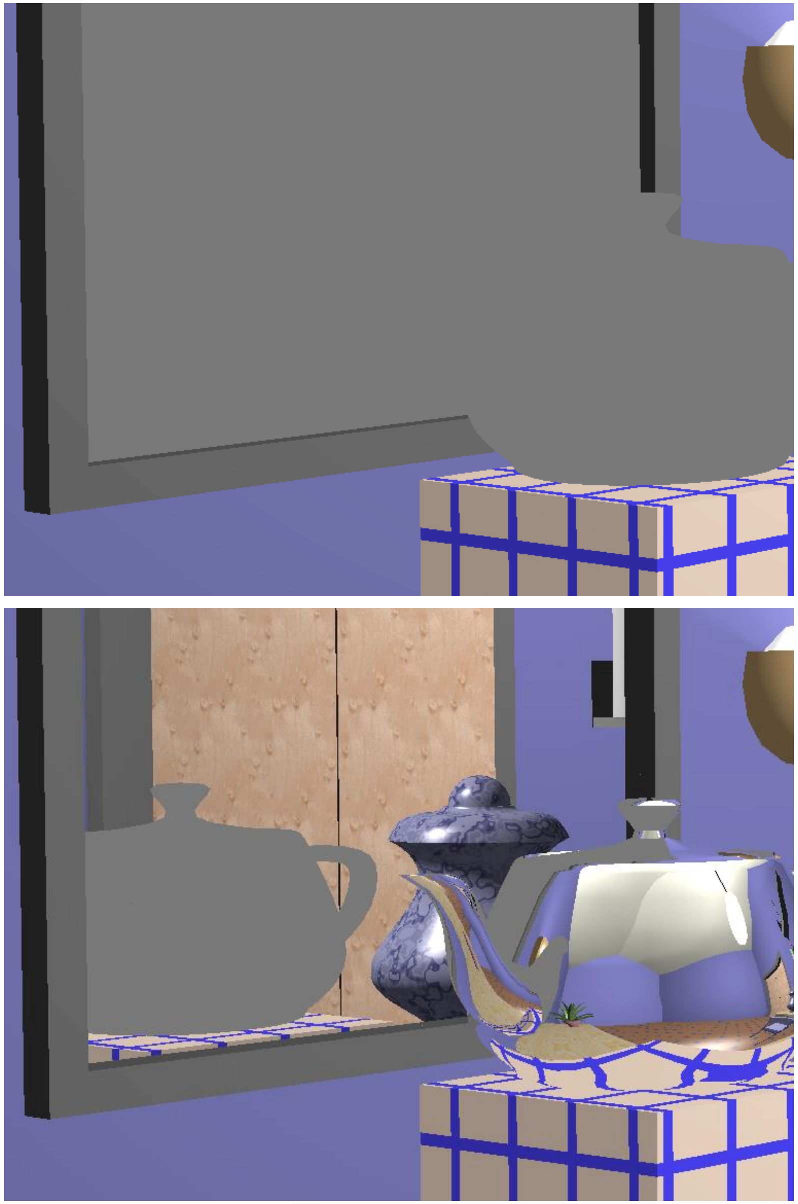


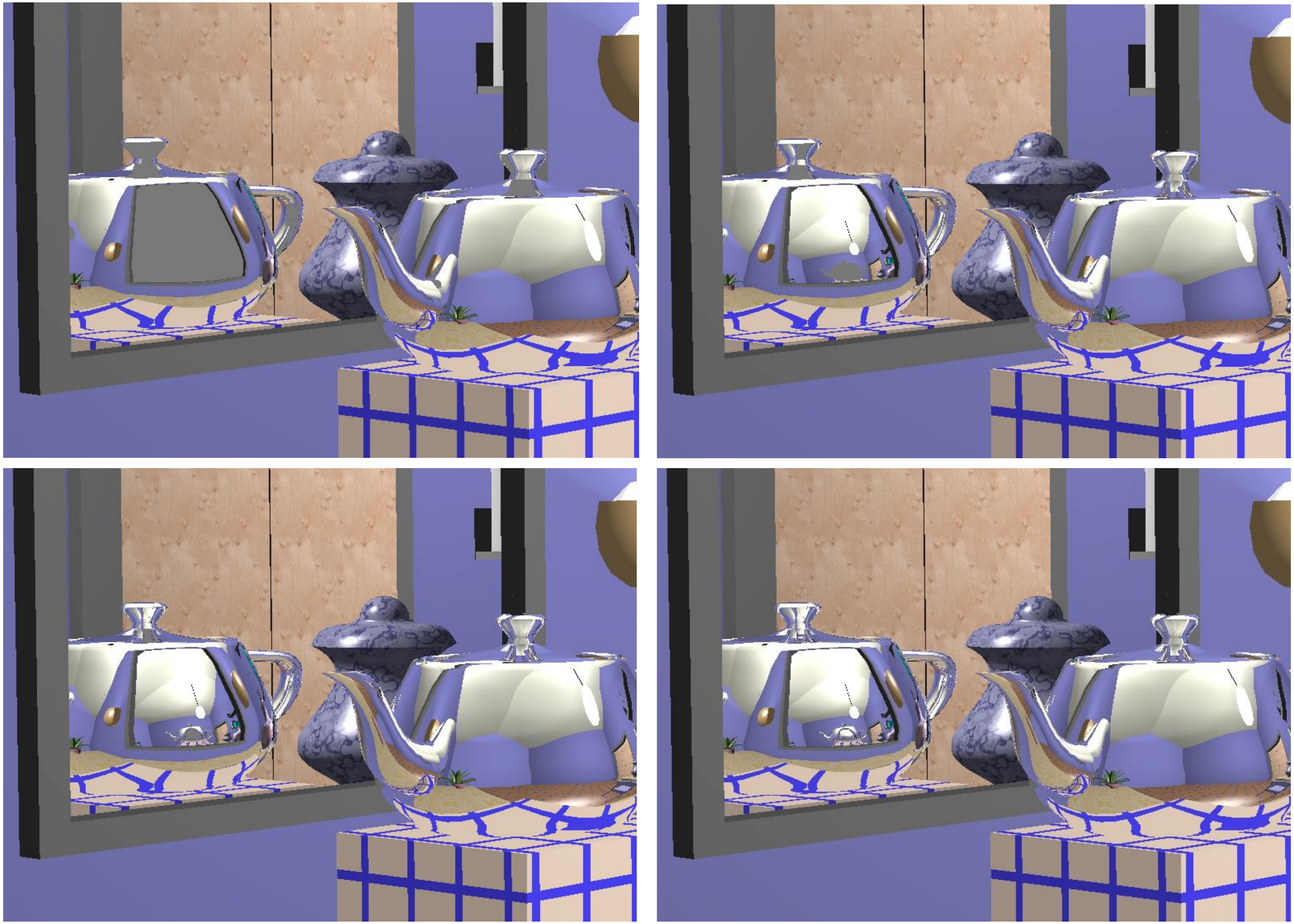
Durand, 06

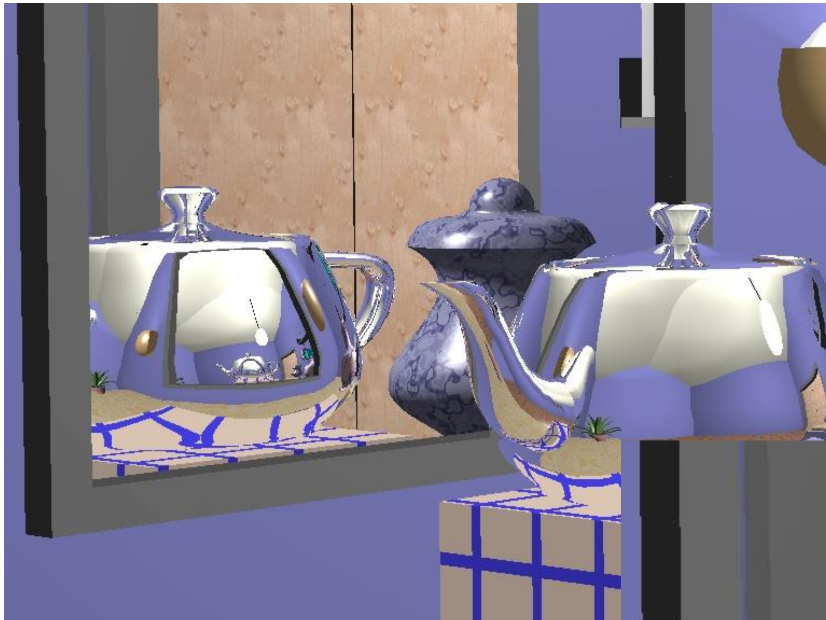
3.7 Recursion termination criteria

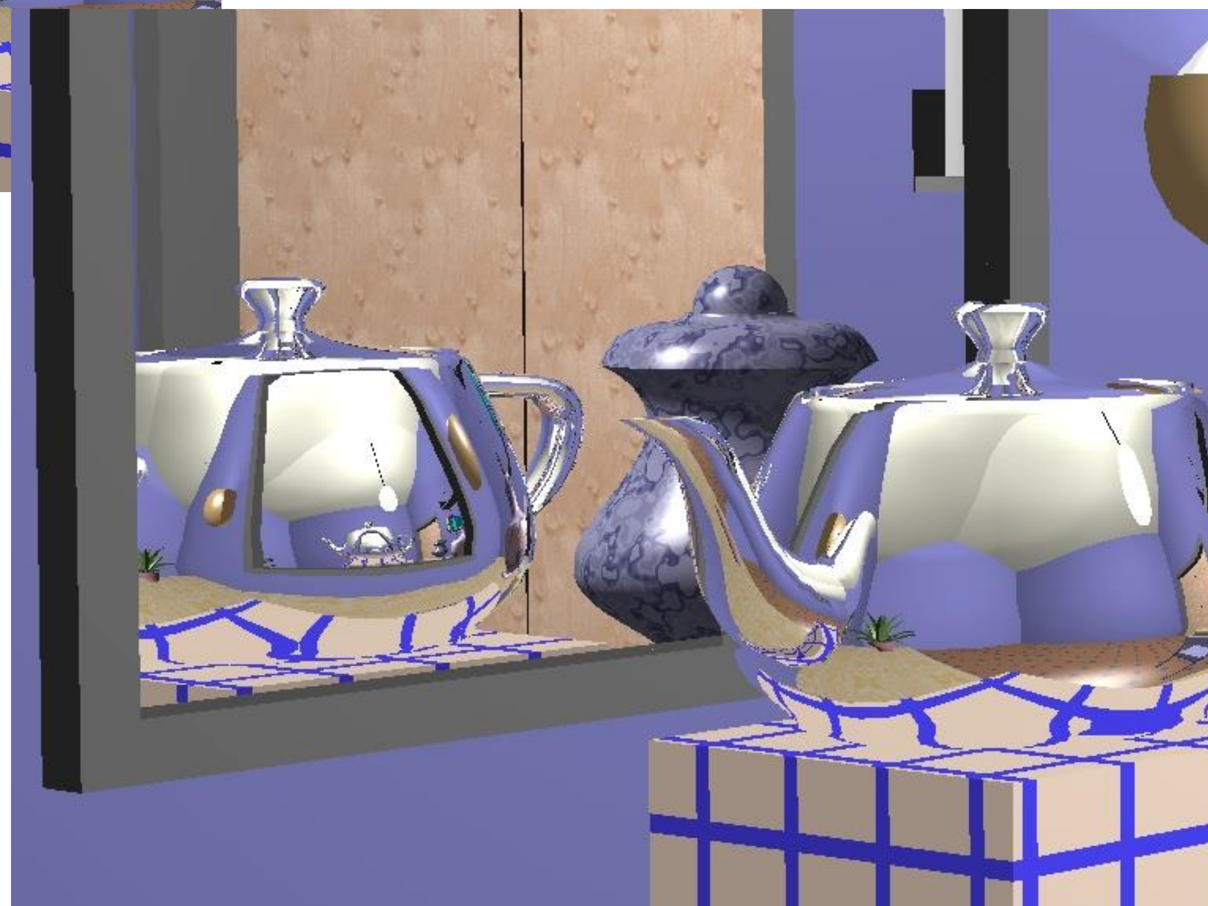
- Terminates if a ray intersects a diffuse surface
- Can terminate when a pre-set depth of recursion has been reached
- Can terminate when the energy of the ray has dropped below a threshold
  - e.g. reflected or transmitted contribution becomes small

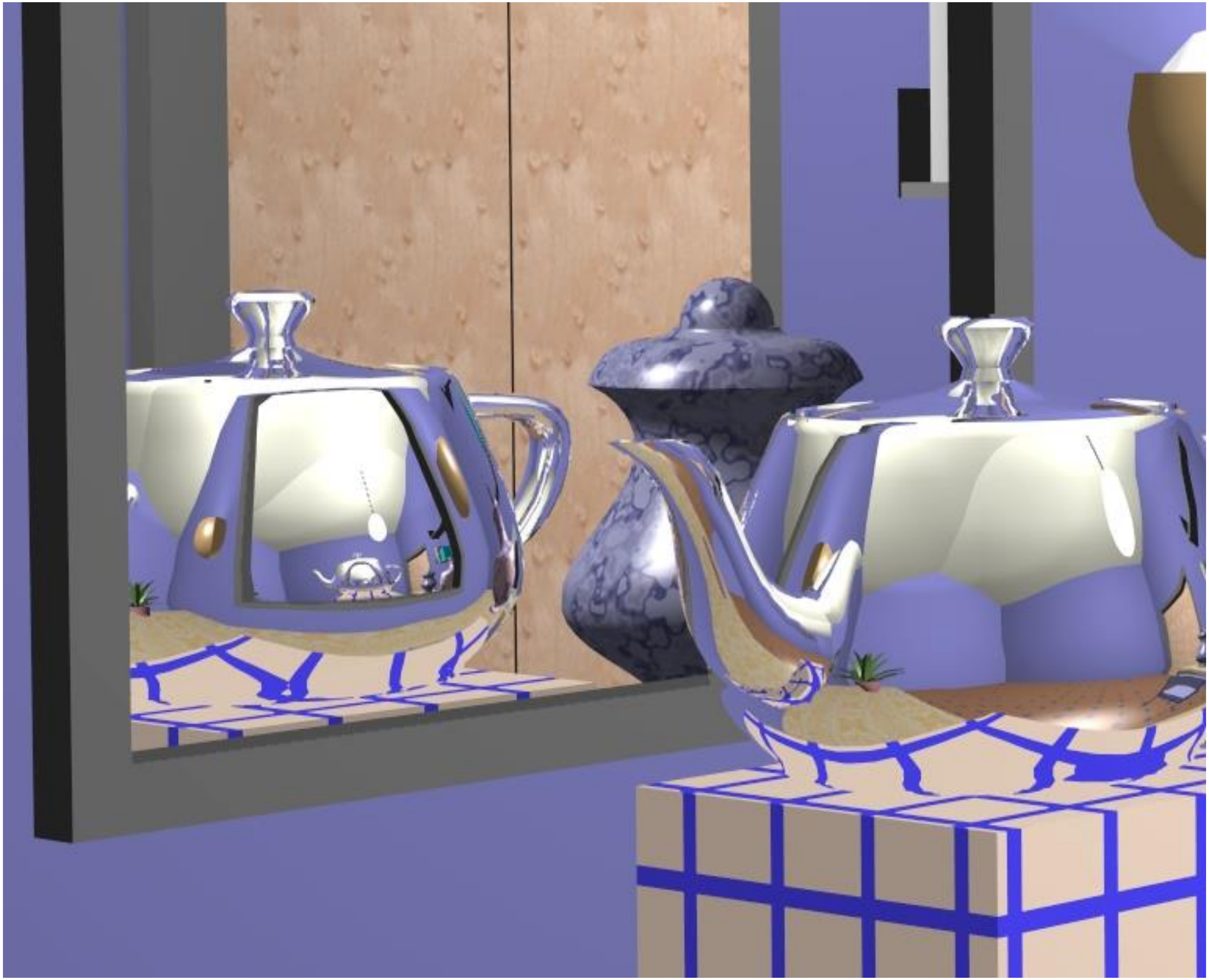
- For recursive depth of zero, rays terminate on the teapot and mirror surface and find no local component. They are (arbitrarily) rendered as grey.
- For recursive depth 1, both the teapot and the mirror find reflections but rays from the eye to the mirror to the teapot terminate at the teapot and produce a grey 'shadow' of the teapot in the mirror.





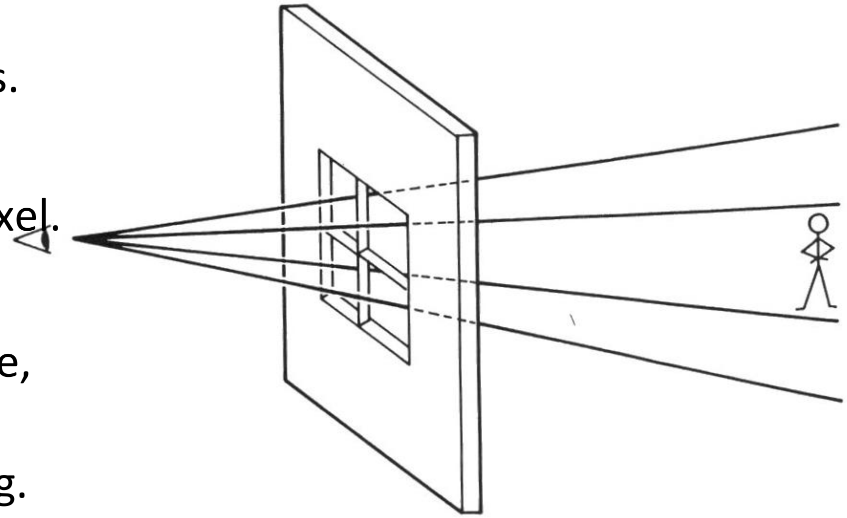






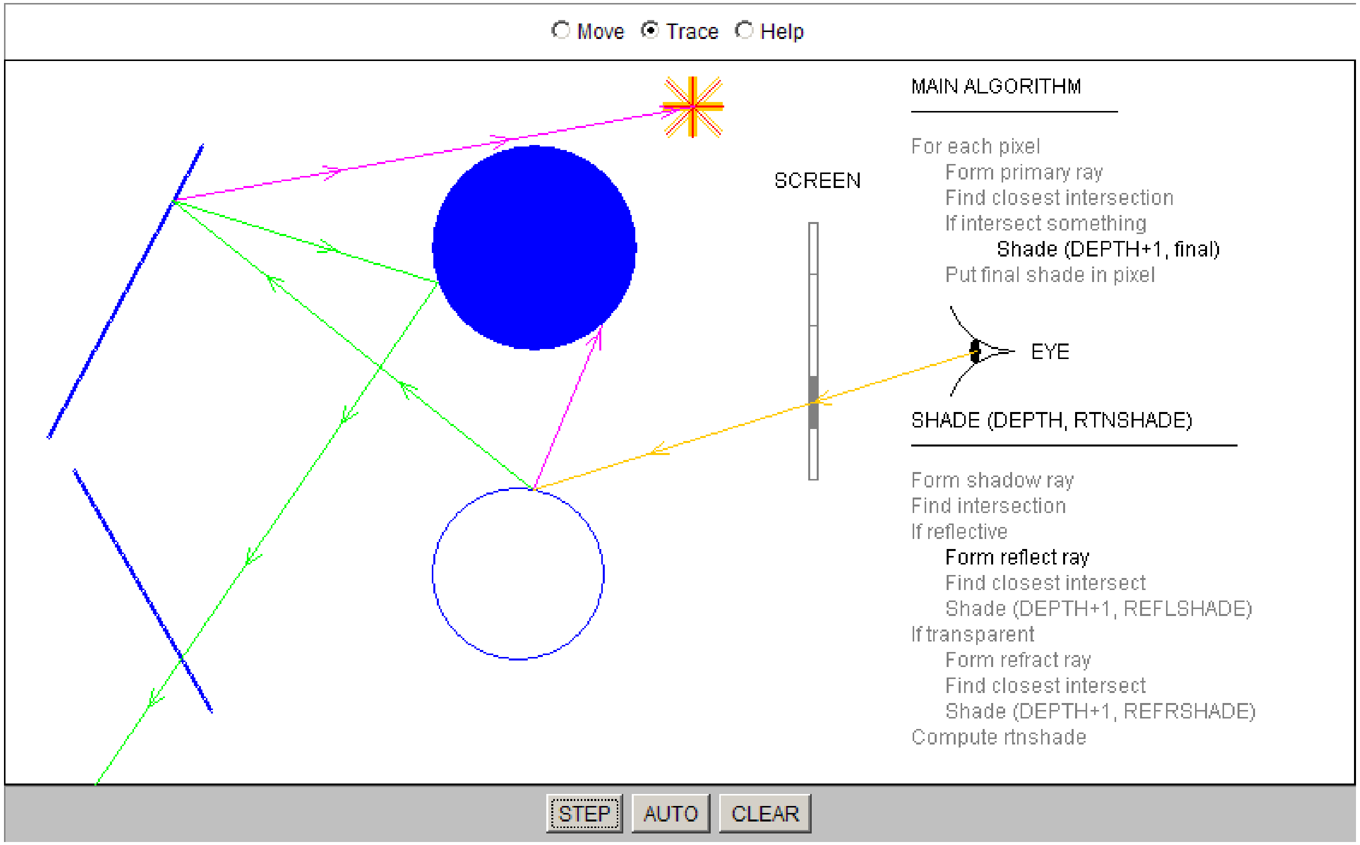
3.8 Issues

- Intersection tests
  - A ray must be tested against all objects
  - What speed-ups can be used?
- Aliasing
  - Sampling leads to aliasing problems.
  - Infinitesimally thin rays.
  - Independent rays – one for each pixel.
- Reality?
  - Photons go from the light to the eye, not the other way!!
  - Full of incorrect approximations, e.g. shadows of transparent objects



3.9 2D Demo

- Written by Yan Liu, under supervision of Dr G. Scott Owen, Georgia State University

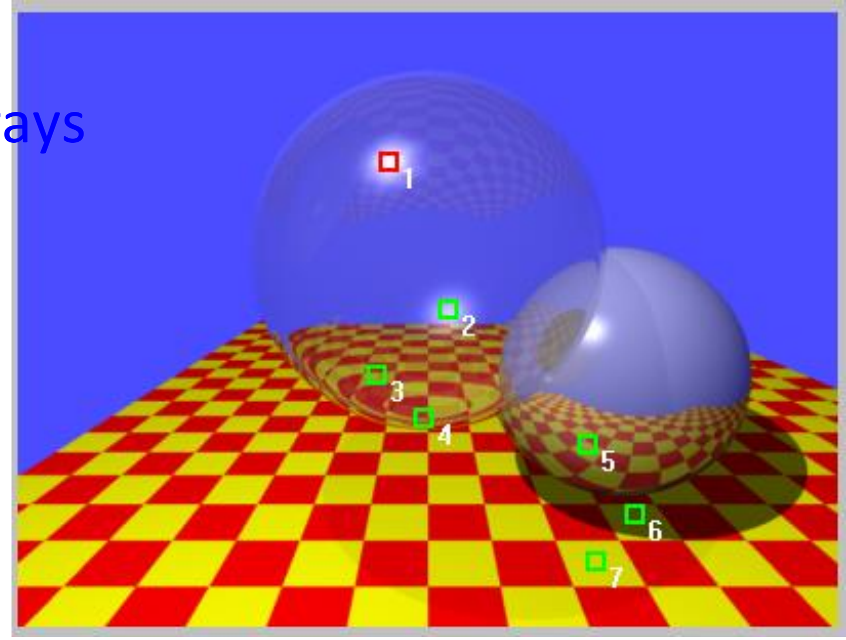


4. Summary

- We trace rays from the eye into the scene
- Naïve ray tracing is impractical since intersection calculations grow exponentially
  - Need speed-ups
- Sampling on a regular grid leads to aliasing
  - Anti-aliasing?
- The global part of the classic (Whitted) algorithm only deals with pure specular-specular interaction
- Direct diffuse (but not diffuse-diffuse) is also covered

- Next lecture:
  - Intersection calculations
  - Speed-up techniques

Epilogue. The adventures of seven rays



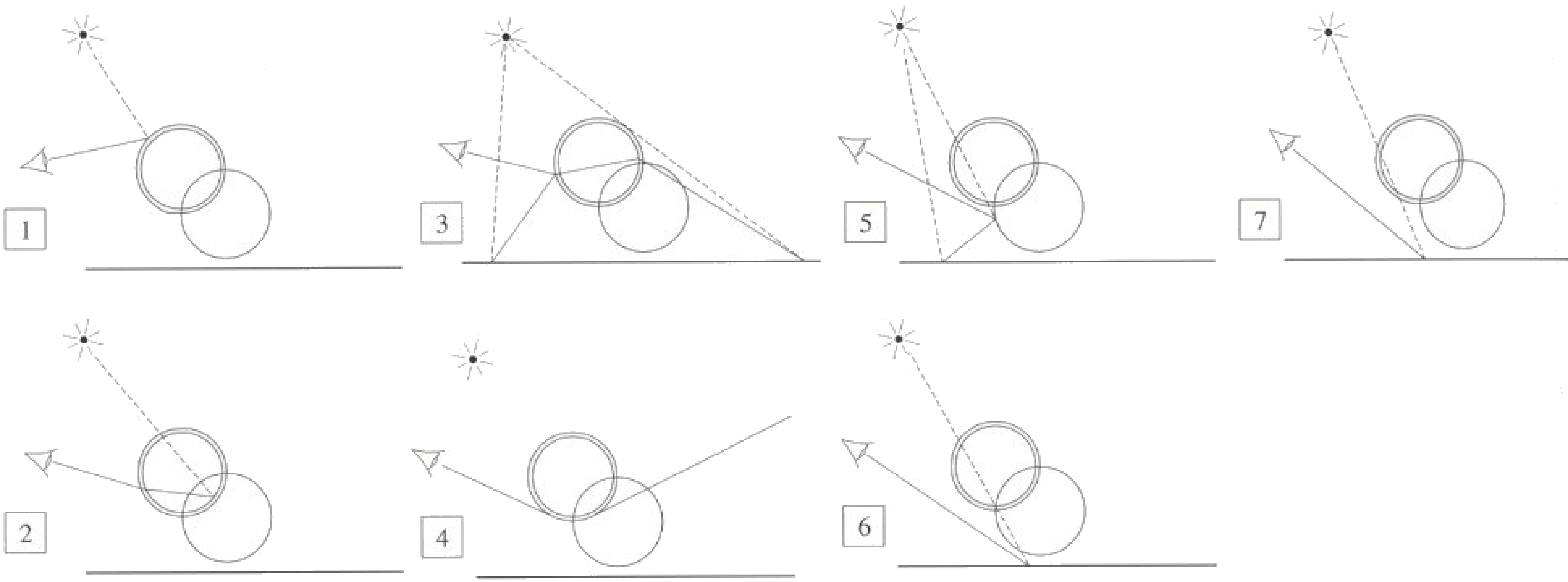
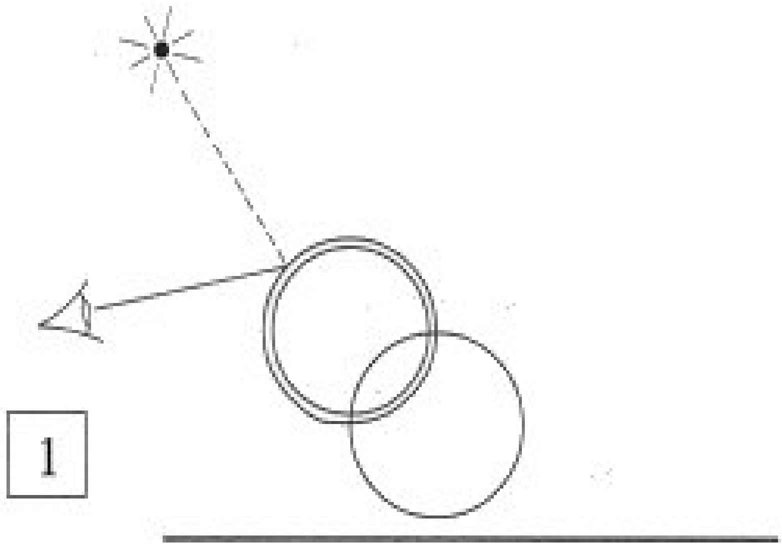


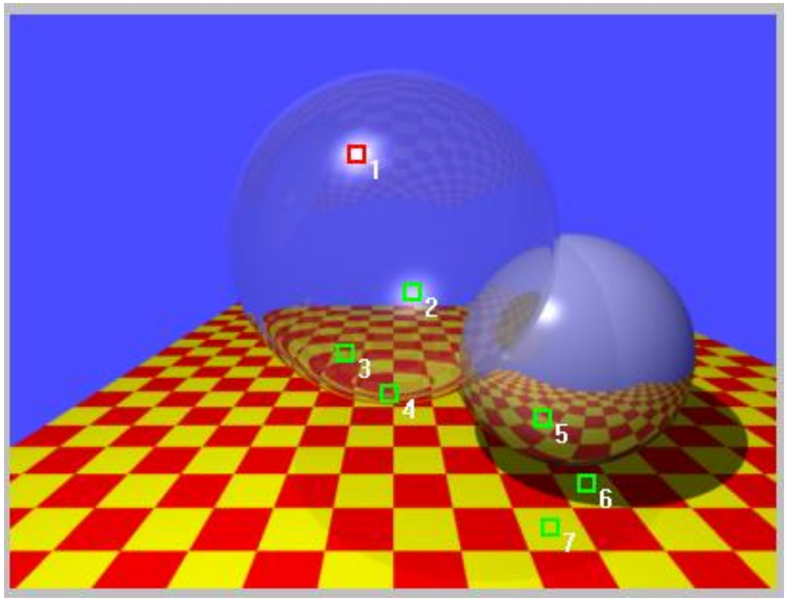
Table 12.1

Very transparent hollow sphere						
k_d (local)	0.1	0.1	0.1	(low)		
k_s (local)	0.8	0.8	0.8	(high)		
k_{rg}	0.1	0.1	0.1	(low)		
k_{tg}	0.9	0.9	0.9	(high)		
Opaque (white) sphere						
k_d (local)	0.2	0.2	0.2	(white)		
k_s (local)	0.8	0.8	0.8	(white)		
k_{rg}	0.4	0.4	0.4	(white)		
k_{tg}	0.0	0.0	0.0			
Chequerboard						
k_d (local)	1.0	0.0	0.0/1.0	1.0	0.0	(high red or yellow)
k_s (local)	0.2	0.2	0.2			
k_{rg}	0					
k_{tg}	0					
Blue background						
k_d (local)	0.1	0.1	1.0	(high blue)		
Ambient light	0.3	0.3	0.3			
Light	0.7	0.7	0.7			

Ray 1

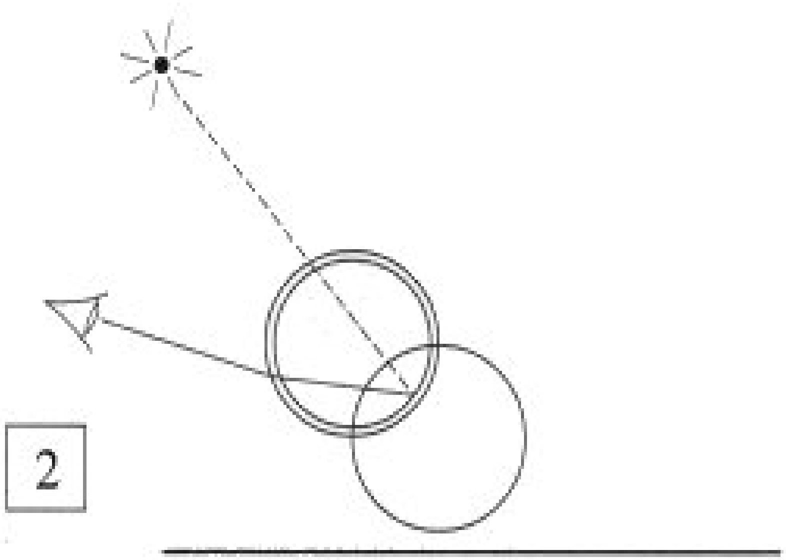
- Because it is near the mirror direction L
  - $I_{local}(P)$ is high
  - $K_{rg}I(P_r)$ is low

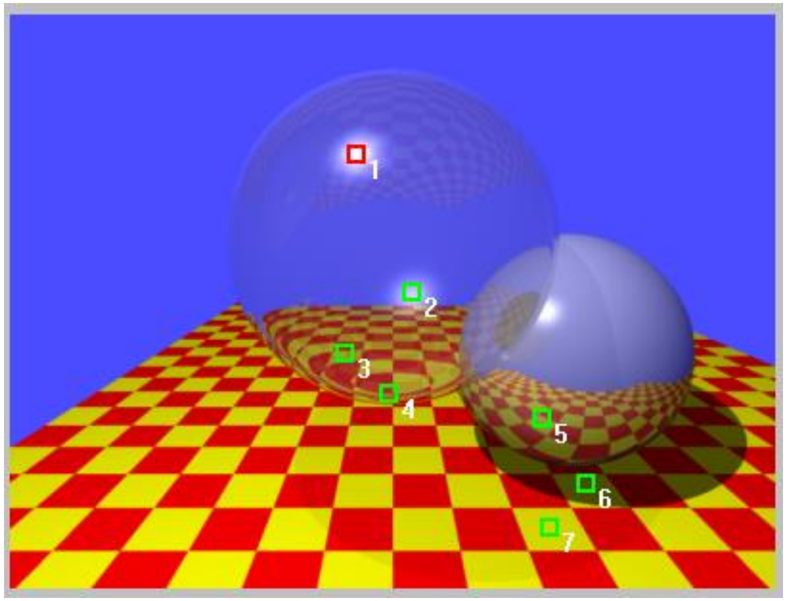




Ray 2

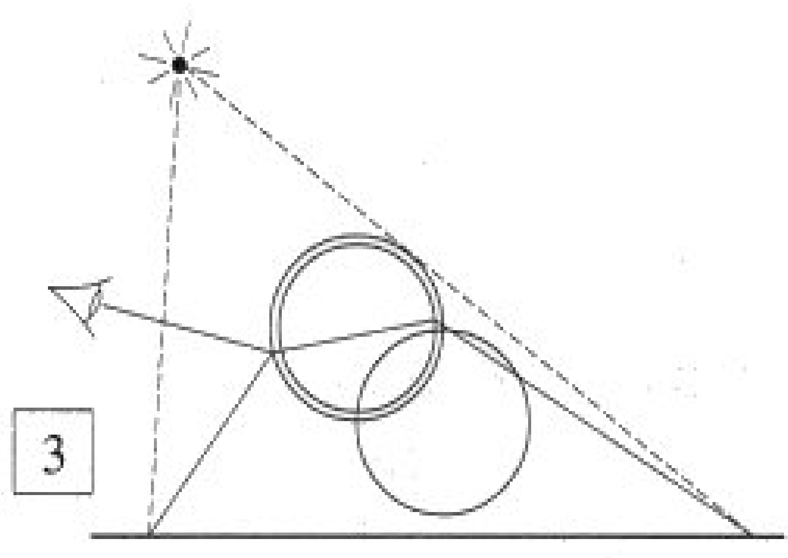
- Similar to ray 1, except specular highlight appears on inside wall of hollow sphere.
- Specular highlight is in wrong position, since no refraction of light ray used to calculate local reflection.

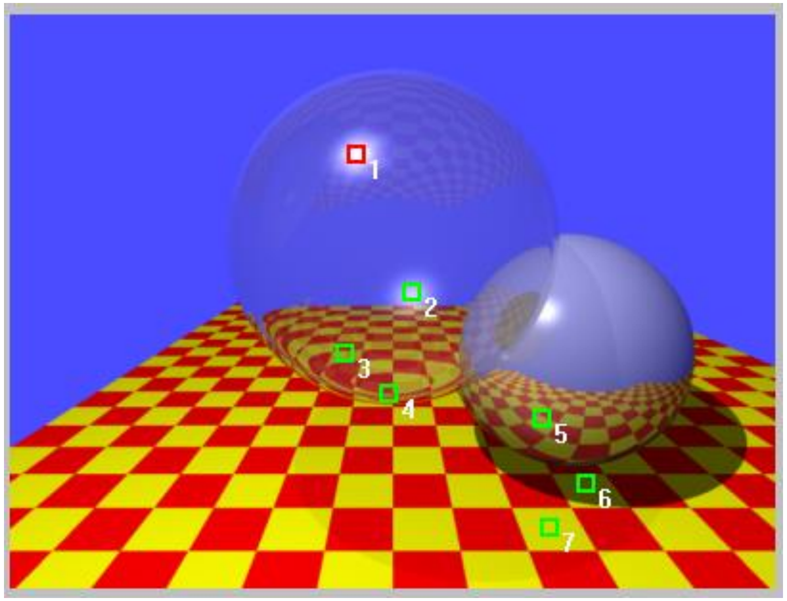




Ray 3

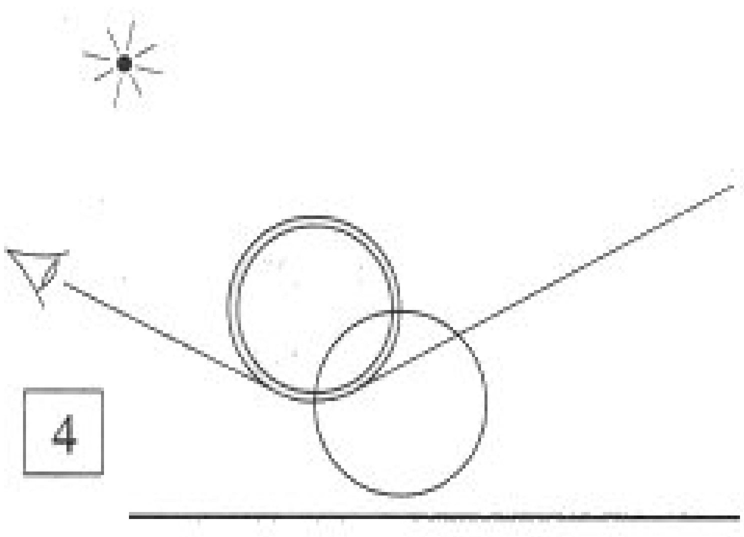
- Local contribution at hollow sphere hits is (almost) zero.
- Chequerboard refracts through the sphere.
- Also reflection on outside of sphere.

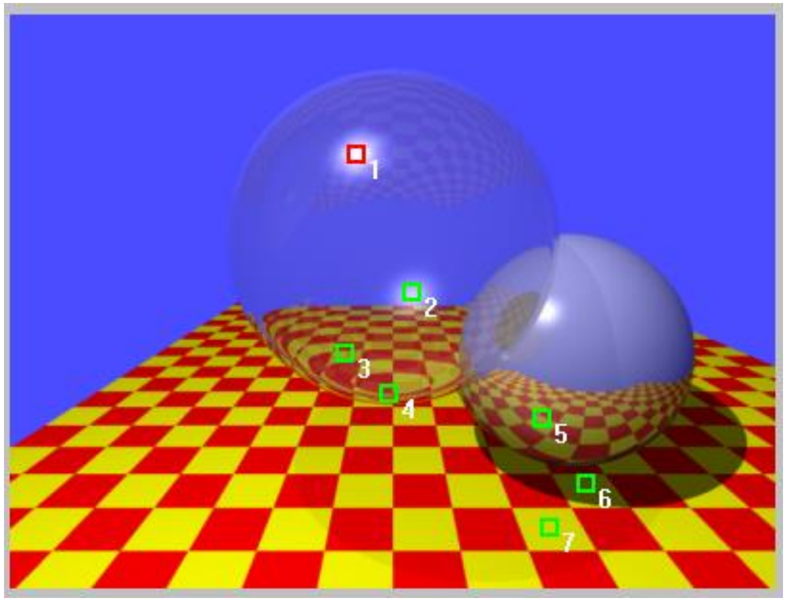




Ray 4

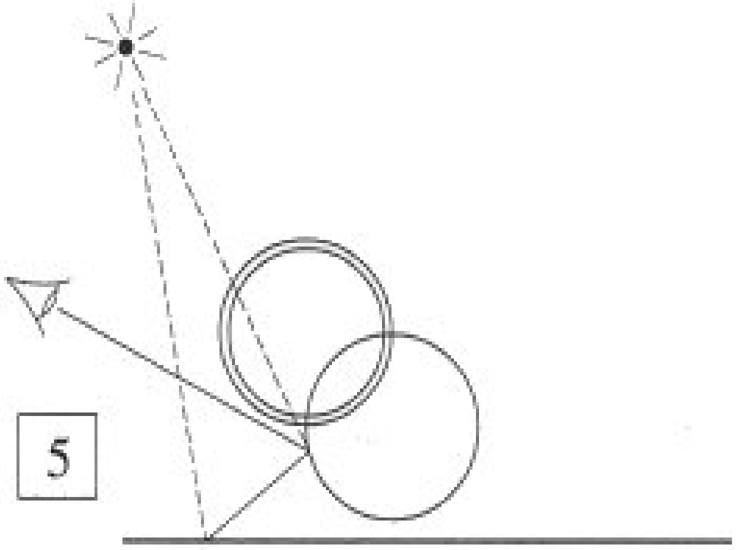
- High refraction gives blue background

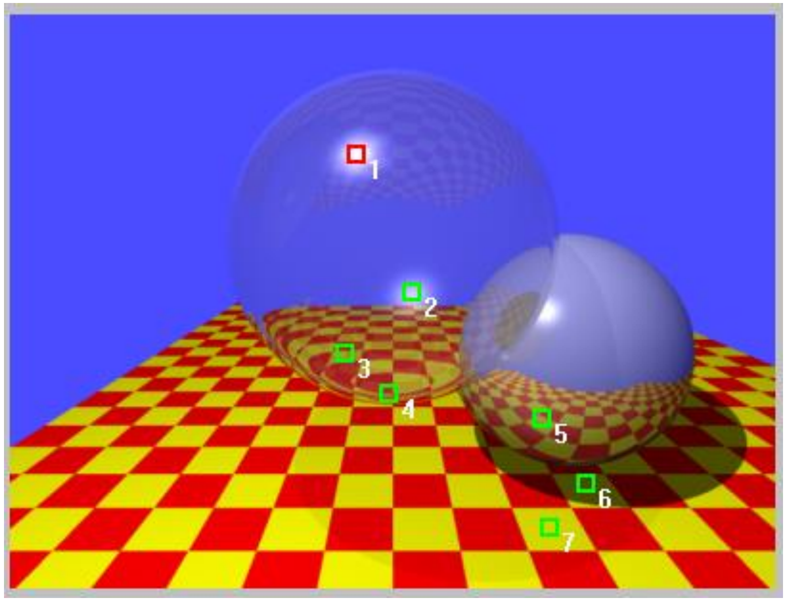




Ray 5

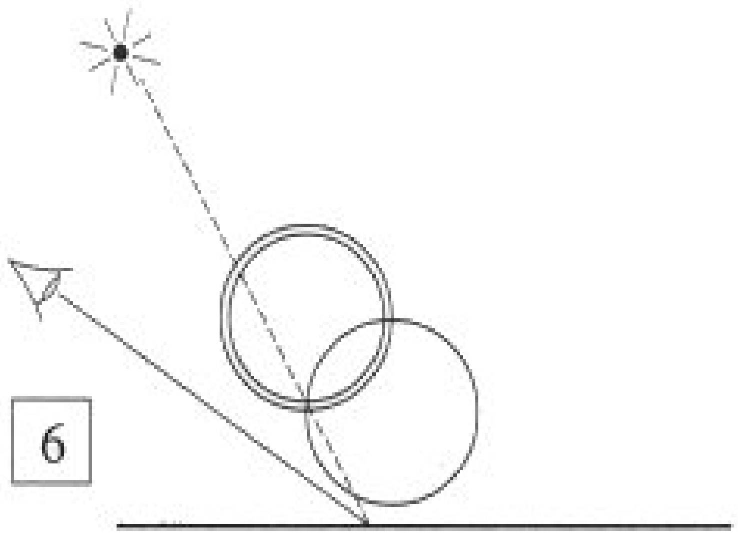
- Local component and chequerboard reflection.

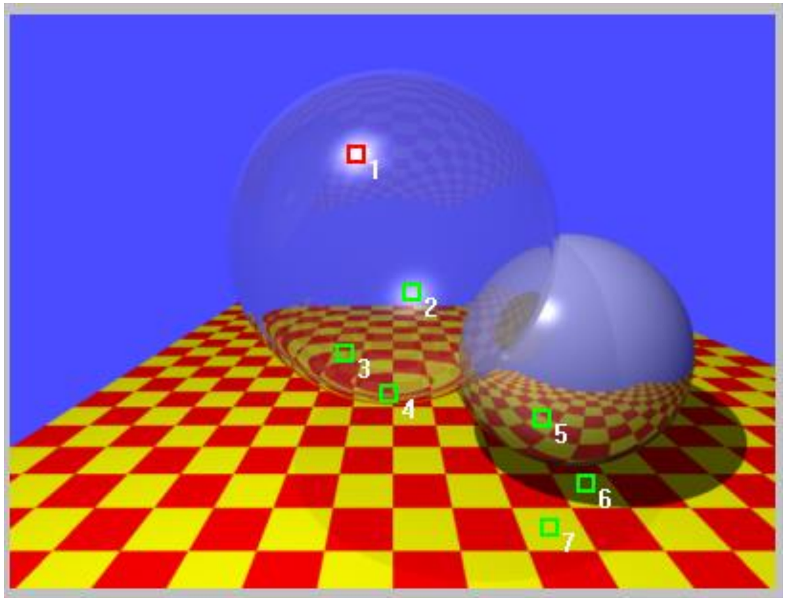




Ray 6

- Local component of chequerboard plus shadow .





Ray 7

- Similar to 6 except shadow is less due to light through hollow sphere.