COMP10001 Foundations of Computing

Semester 2 2014

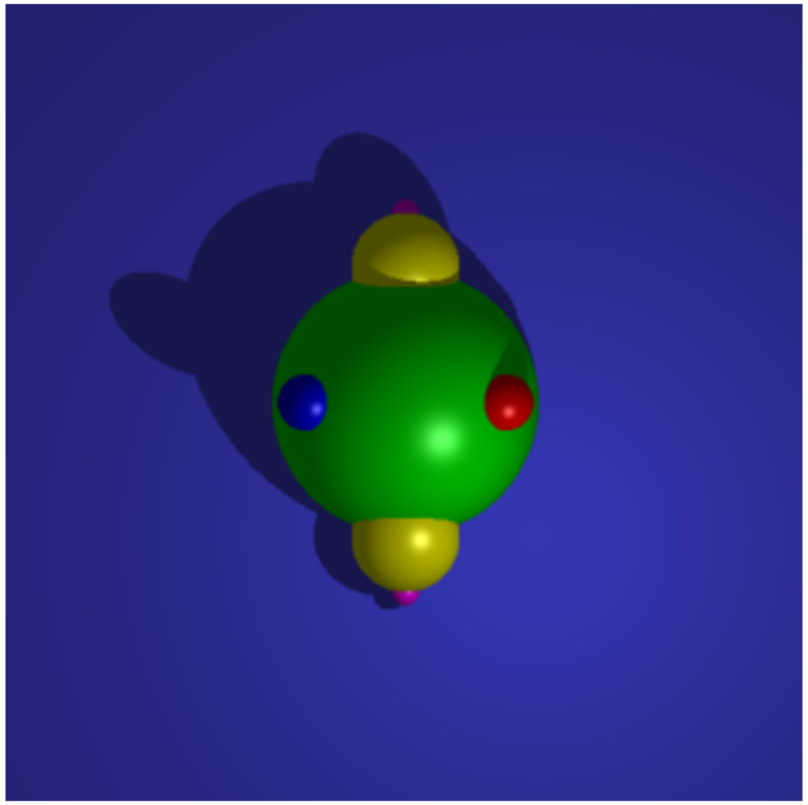
Lecture 18 (advanced lecture, not examinable)

Ray Tracing

Bernie Pope, bjpope@unimelb.edu.au

Outline

- What is ray tracing?
- A ray tracing program in Python, ray.py
- An illumination model
- Ray-object intersection
- Recursive ray tracing
- Optimisation



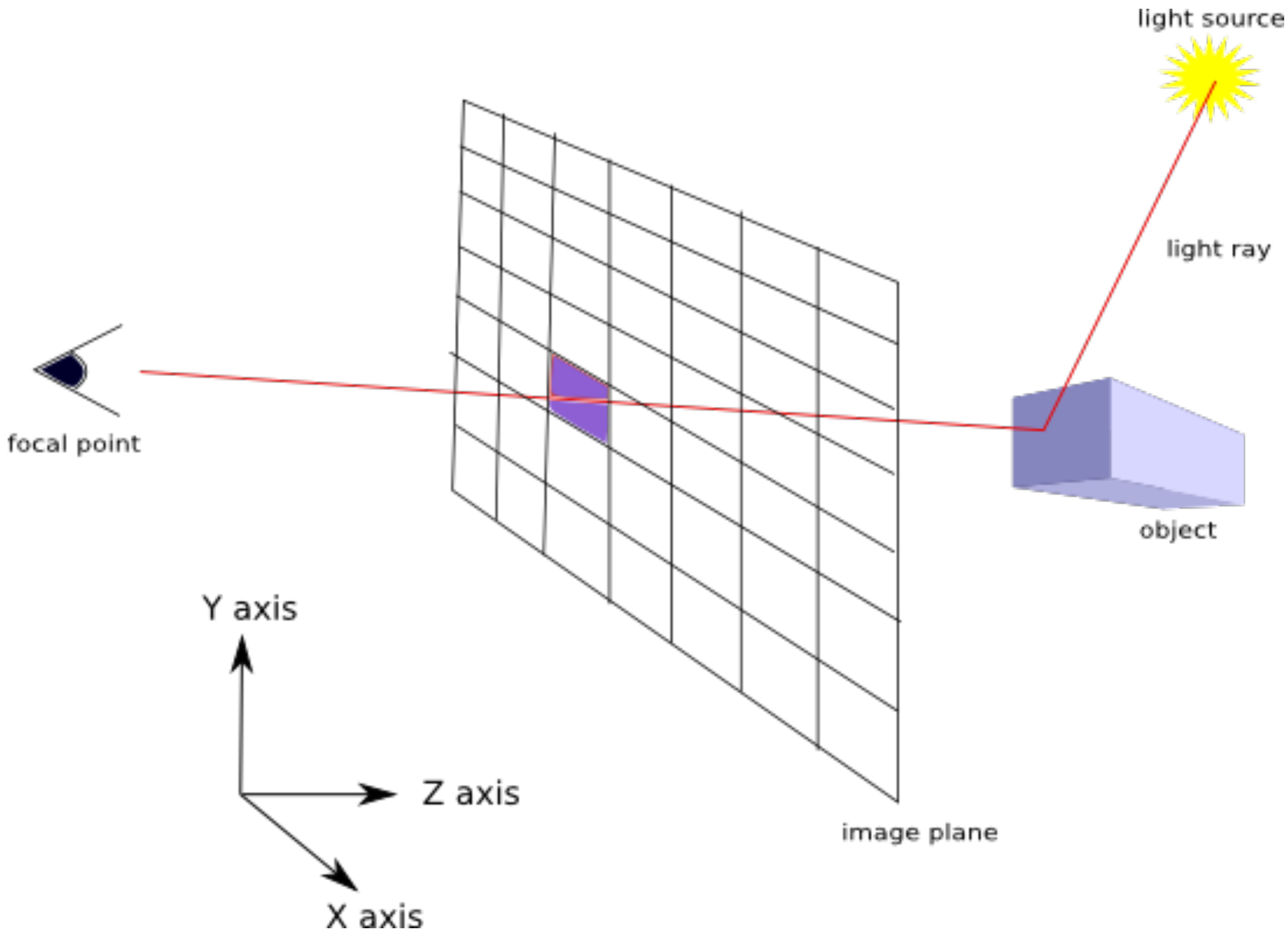
What is ray tracing?

- A technique for generating 3D computer graphics.
- Based on a simple model of light rays on surface properties.
- Largely an exercise in geometry.

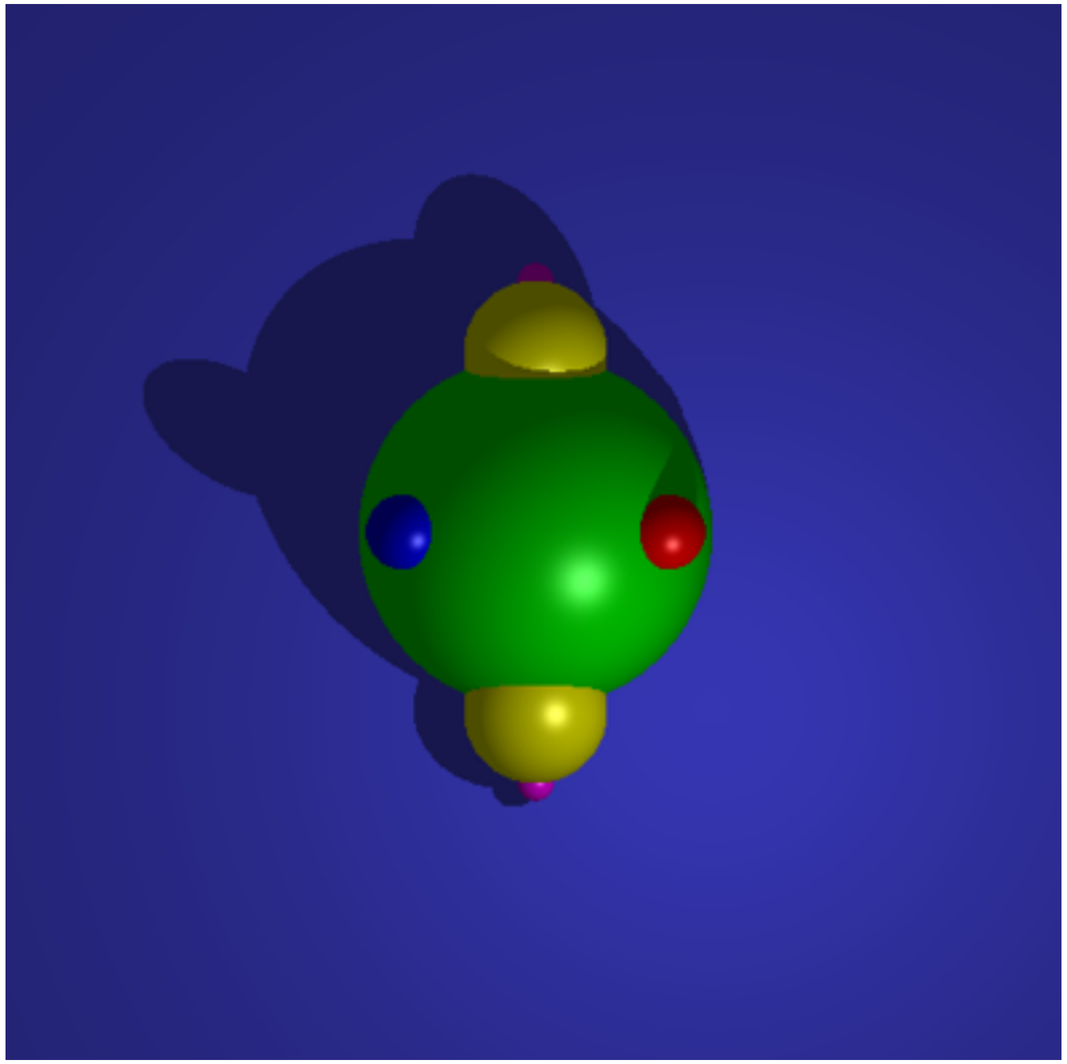
Number one rule of computer graphics

- If it looks good, it is good.

What is ray tracing?

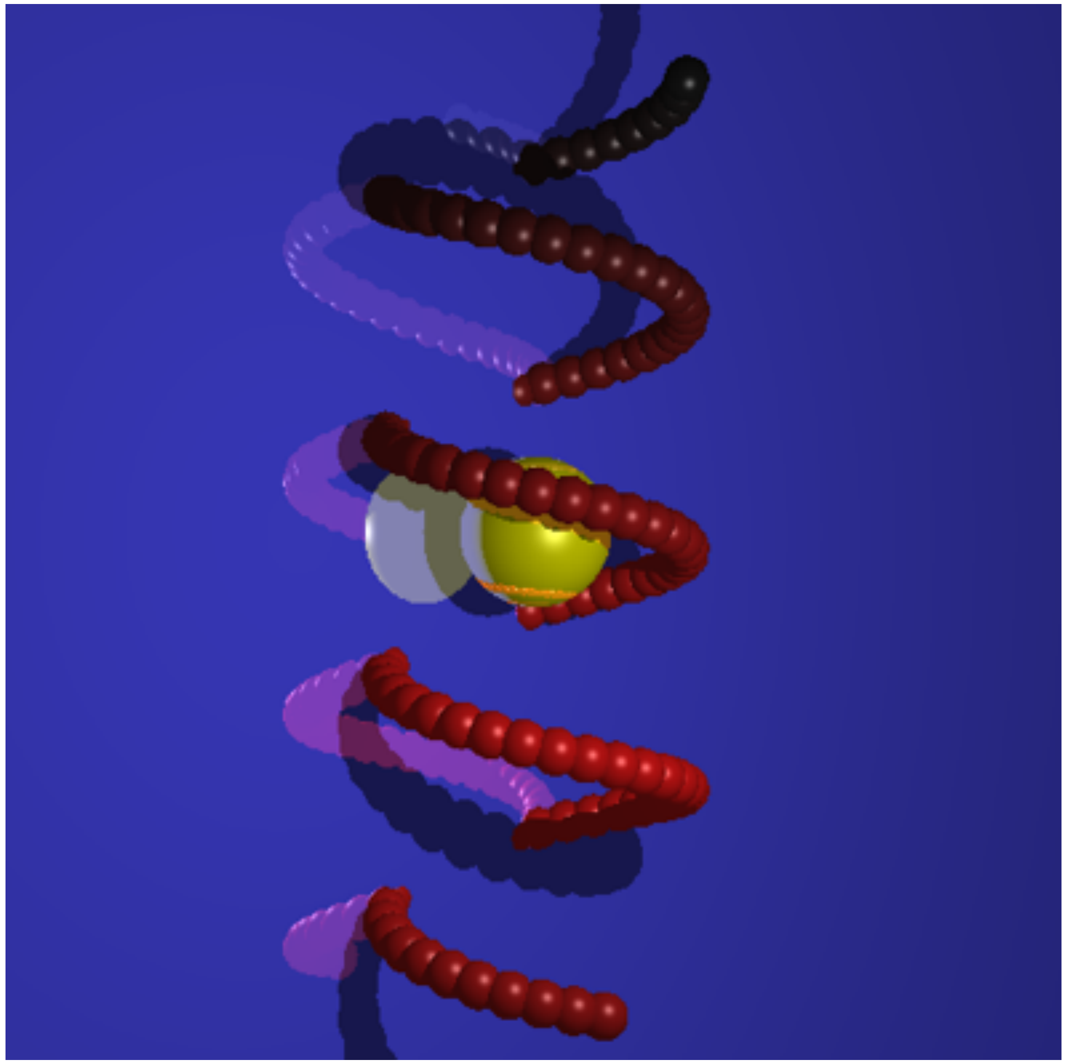


An example image



Generated by ray.py, code available on LMS

Another example image



Generated by ray.py, code available on LMS

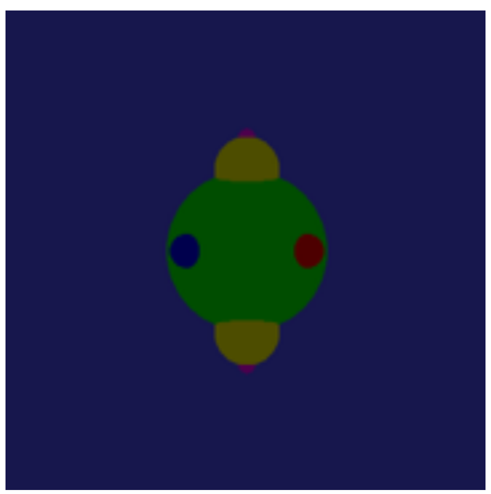
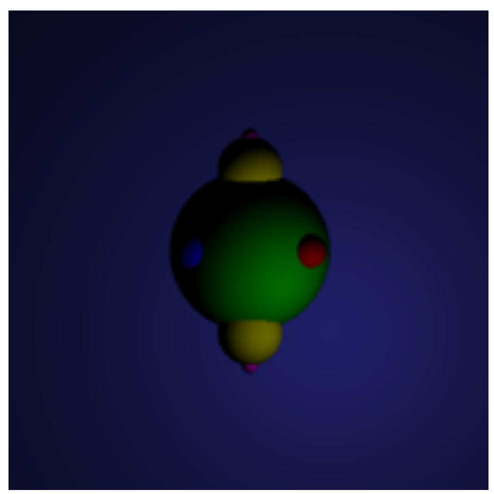
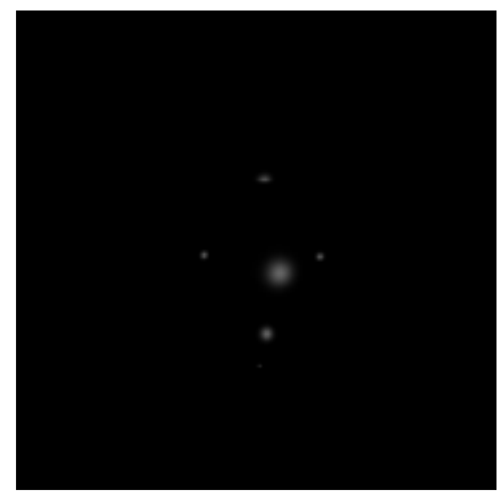
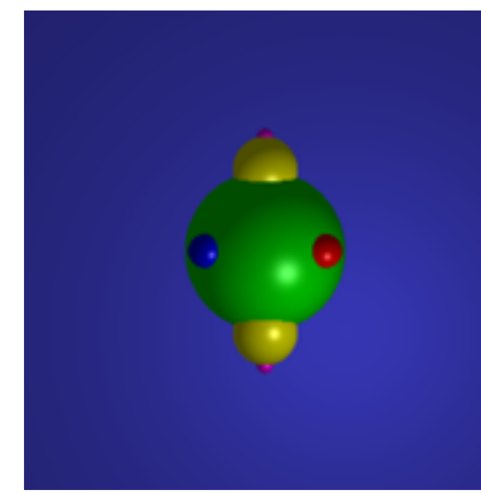
Ray tracing algorithm

```
for x in image_x_coords:
    for y in image_y_coords:
        ray = line(focal, (x, y))
        hits = intersect(ray, objects)
        object = closest(hits, focal)
        pixel = shade(object, lights, focal)
        image[x][y] = pixel
```

ray.py

- I've written a simple ray tracer in Python for you to play with.
- You can download the code from LMS.
- You need to run it on your own computer (not IVLE).
- Let's see how it runs...

The Phong illumination model

Ambient		Diffuse		Specular		Final image
	+		+		=	

Reflections and shadows are extra work on top of this.

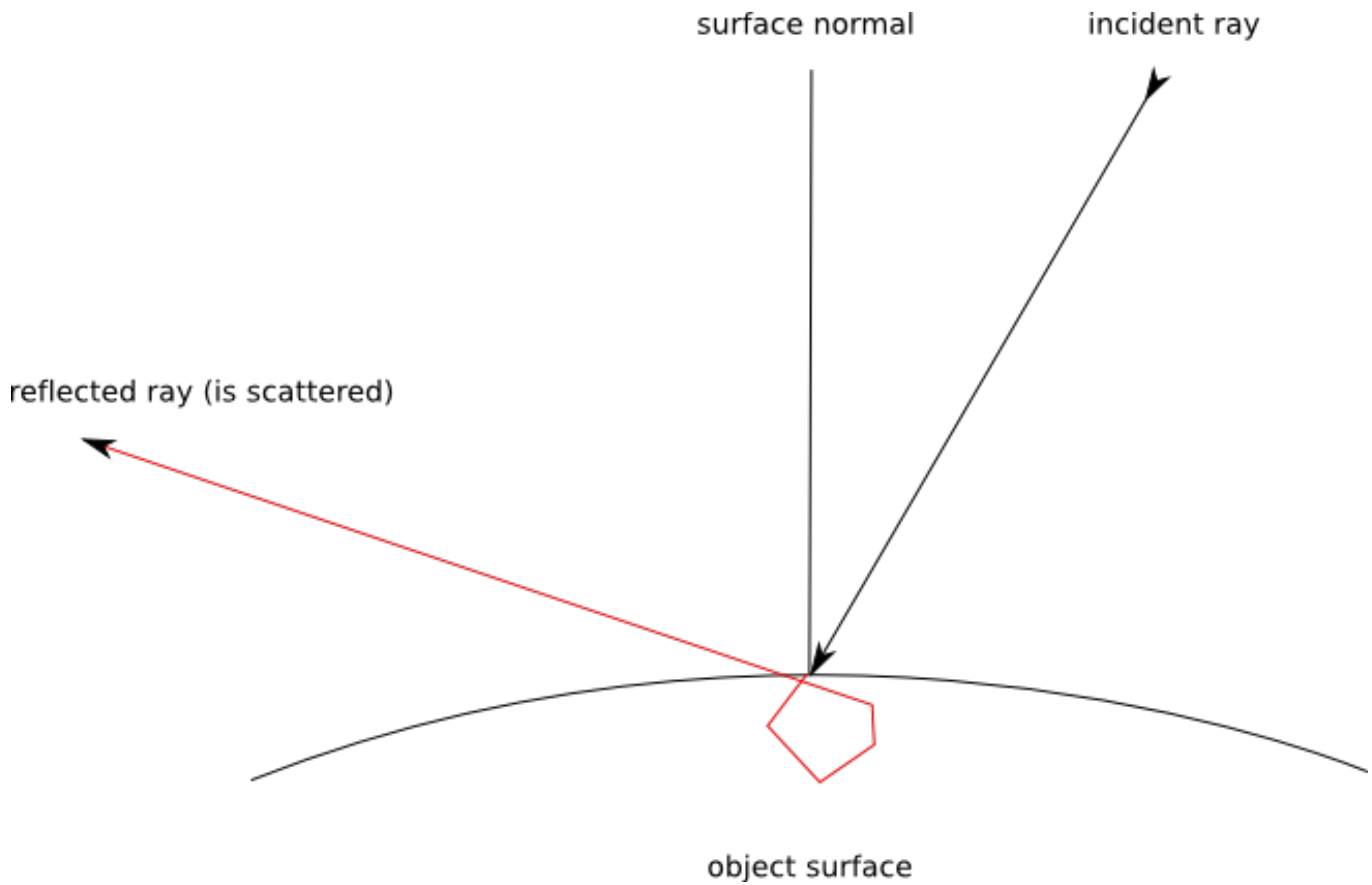
Ambient illumination

- Ambient illumination models the background scattered light in the scene.
- It is a constant.
- It depends on the colour properties of the surface.

Diffuse illumination

- Diffuse illumination models the scattering of light which interacts with the surface of an object.
- It depends on:
  - The colour of the light source. The colour of the surface.
  - The orientation of the surface relative to the direction of the light source.

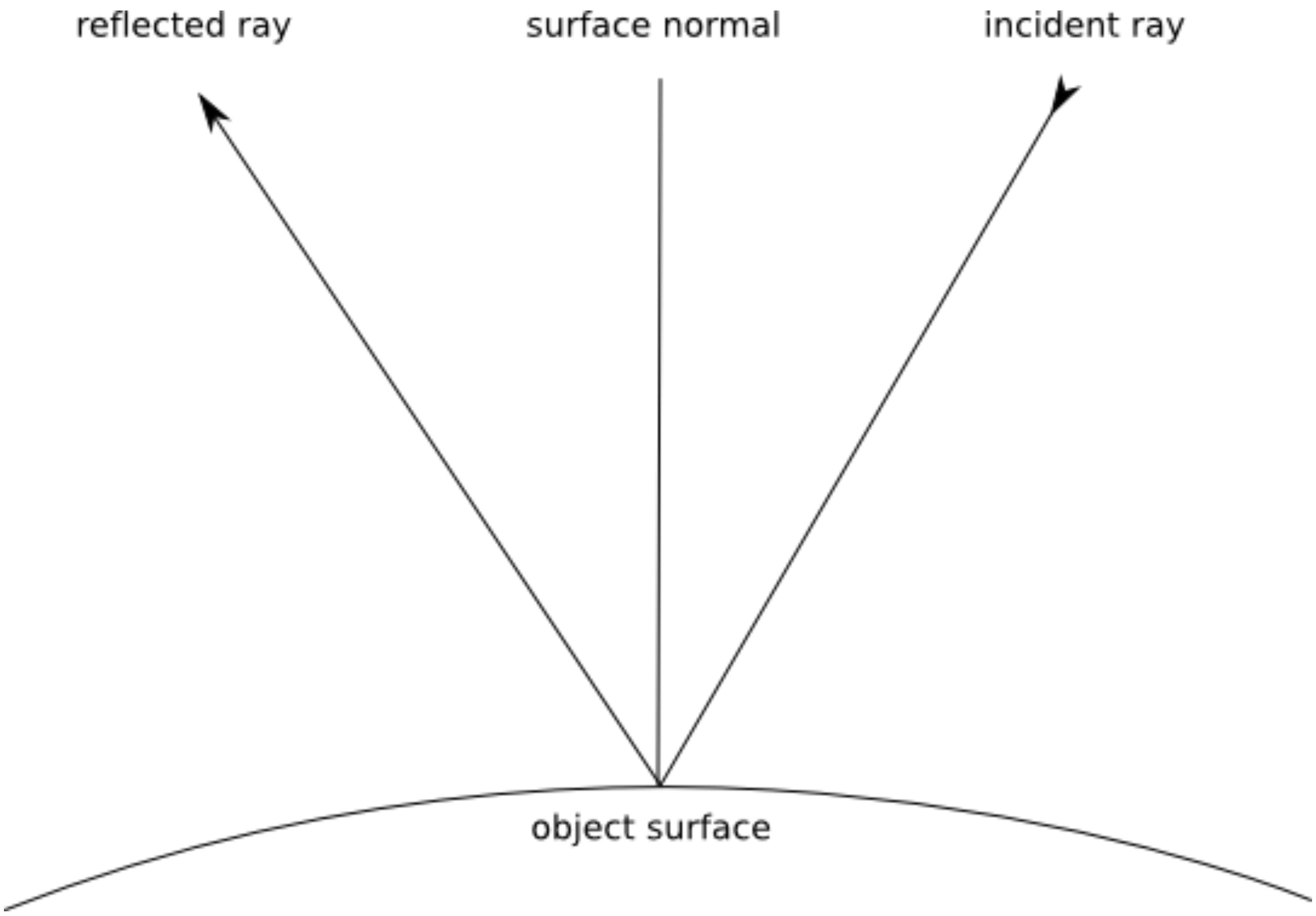
Diffuse illumination



Specular illumination

- Specular illumination models the reflection of the light on a shiny surface.
- It depends on:
    - The colour of the light source.
    - The orientation of the surface relative to the direction of the light source and the viewer.

Specular illumination



Intersection of a ray with a sphere

- Ray through (x_1, y_1, z_1) (x_2, y_2, z_2) defined as:
  - $x = x_1 + u(x_2 - x_1)$
  - $y = y_1 + u(y_2 - y_1)$
  - $z = z_1 + u(z_2 - z_1)$
- Sphere with radius r and center (x_3, y_3, z_3) defined as:
  - $(x - x_3)^2 + (y - y_3)^2 + (z - z_3)^2 = r^2$

Intersection of a ray with a sphere

- Substitute the ray equations for x, y, z into the sphere equation.
- We end up with a quadratic equation of the form:

  $au^2 + bu + c = 0$

- We solve for the parameter u.
- There can be zero, one or two solutions.
- u is the distance of the intersection along the ray from

(x_1, y_1, z_1).

Recursive ray tracing

- For rendering:
    - Mirrored surfaces.
    - Translucent surfaces.
- At the point on an object surface intersected by a ray, compute its reflected and/or refracted ray.
- Apply the same rendering algorithm as before for the new ray, taking the point on the object to be the new viewer position.

Optimisation

- Ray tracing is *pleasantly parallel*: you can make it go fast by rendering pixels (or other parts of an image) at the same time on a parallel computer.
- Individual pixel calculations can be sped up by avoiding unnecessary ray-object intersection calculations - usually done by *space partitioning*.
- Hardware accelerators (GPUs) can speed up geometric calculations.

Other 3D techniques

- Rasterization: used in computer games, fast, not physically realistic.
- Photon mapping: computationally expensive, but realistic results, especially with non-direct illumination effects, such as caustics.