Lab #3 - Ray tracing (part 1)

Informática Gráfica

Adolfo Muñoz - Julio Marco
Pablo Luesia - J. Daniel Subías – Óscar Pueyo

Before we begin…

Graphics and Imaging Lab

- Requirements for this lab:
  - Points, directions, matrices and their operations

Before we begin…

Graphics and Imaging Lab

- Requirements for this lab:
  - Points, directions, matrices and their operations
- Careful with the submissions
  - Lab 3 (ray tracing) does not need to be submitted
    - Recommended deadline: **October 18th**

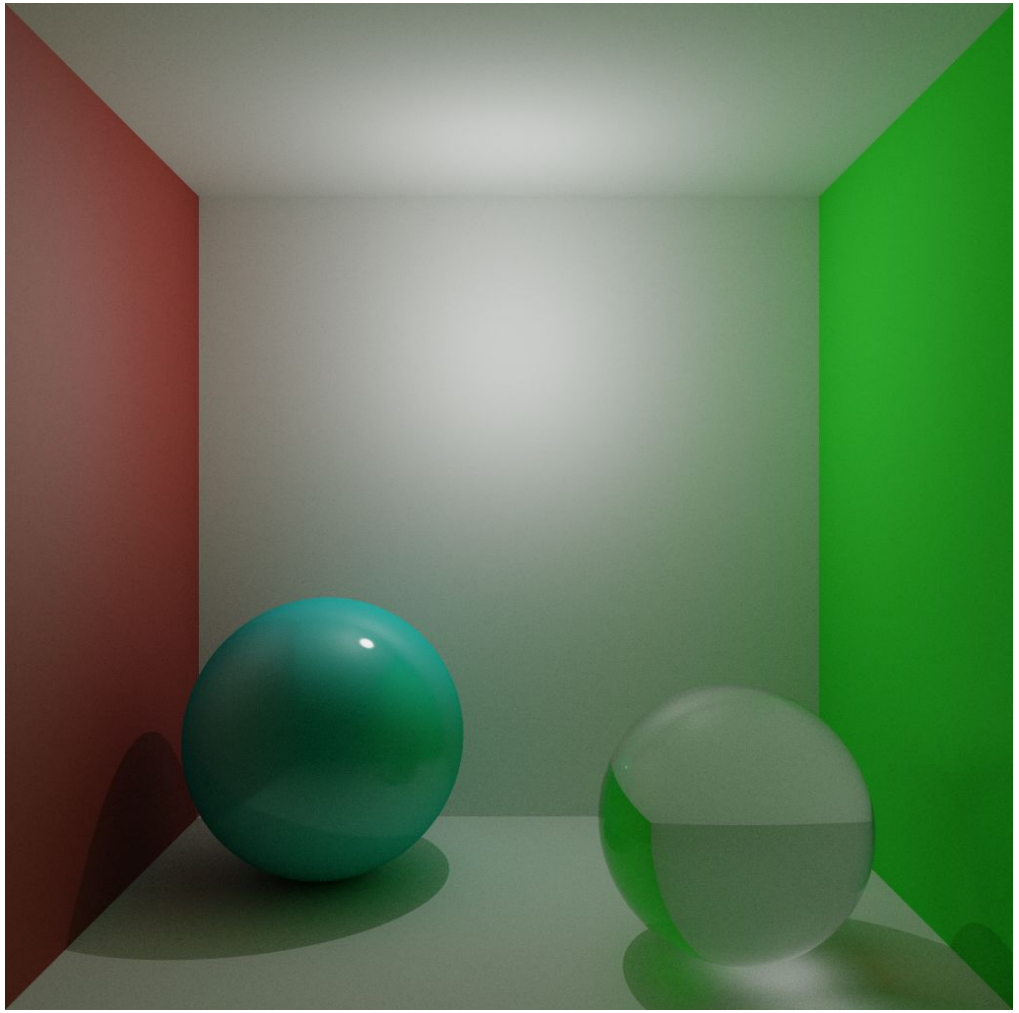
Before we begin…

Graphics and Imaging Lab

- Requirements for this lab:
  - Points, directions, matrices and their operations
- Careful with the submissions
  - Lab 3 (ray tracing) does not need to be submitted
    - Recommended deadline: **October 18th**
  - Lab 4 (path tracing) **will be submitted** at the end of the course
    - All of the code you write today will be used for Lab 4
    - Recommended deadline: November 13th (moodle: January 11th, this is only a recommendation)
- Remember: Final work is 80% of the final grade

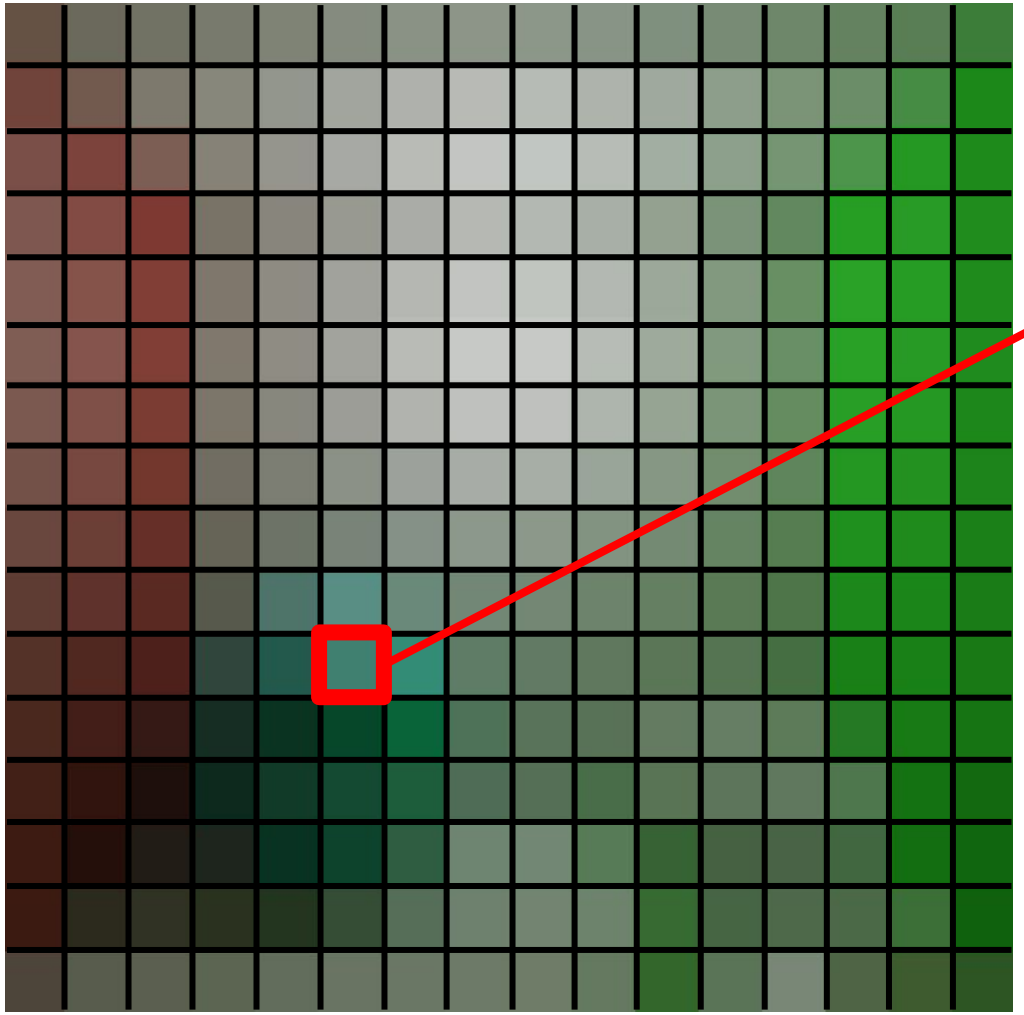
Which color do we fill each pixel with?

Graphics and Imaging Lab



Which color do we fill each pixel with?

Graphics and Imaging Lab

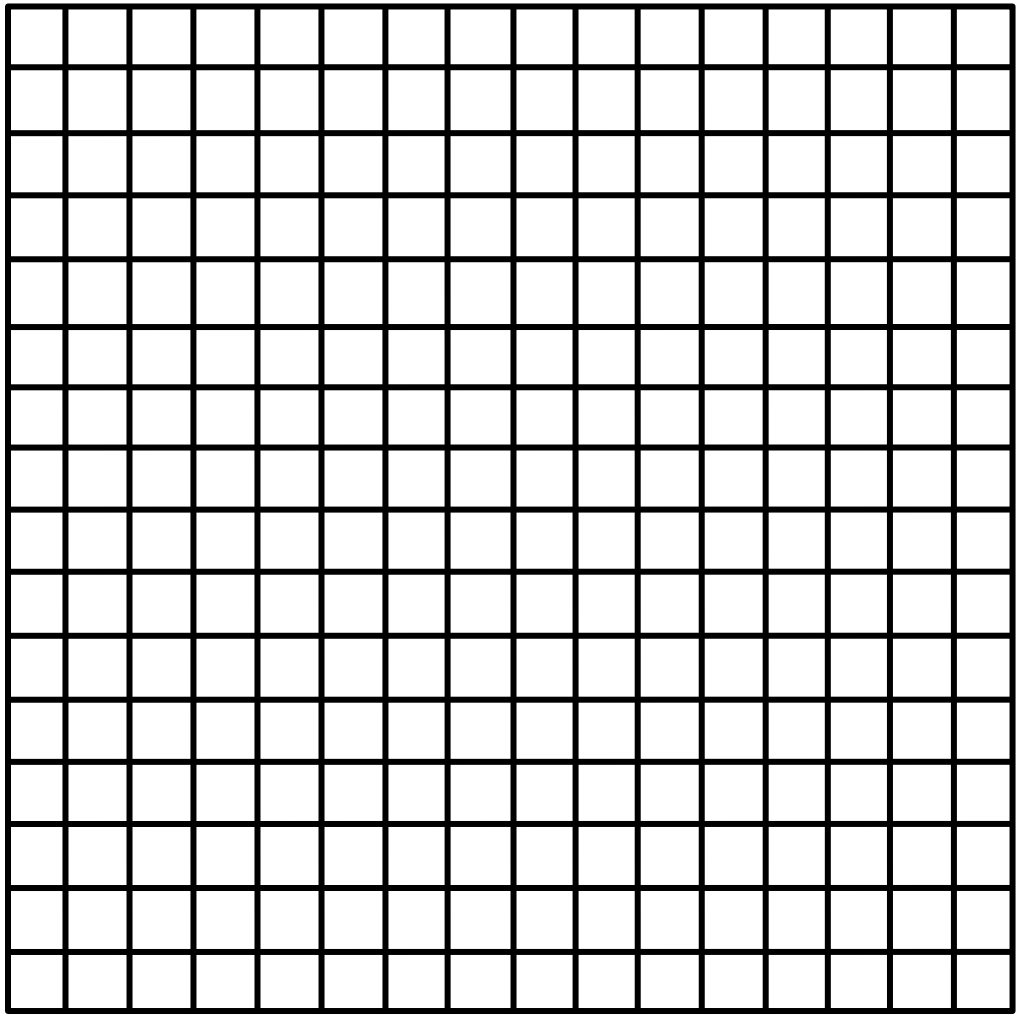


R = 0.25 G = 0.5 B = 0.45

But how?

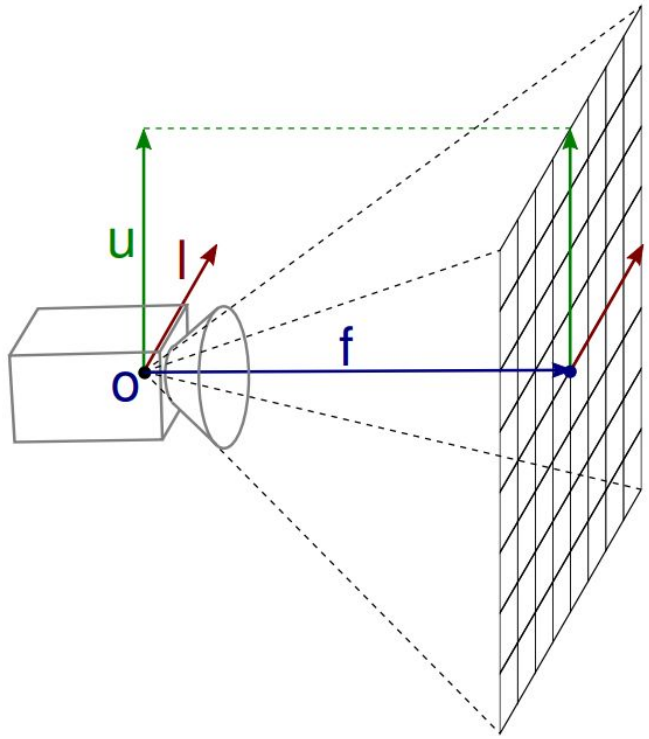
Which color do we fill each pixel with?

Graphics and Imaging Lab



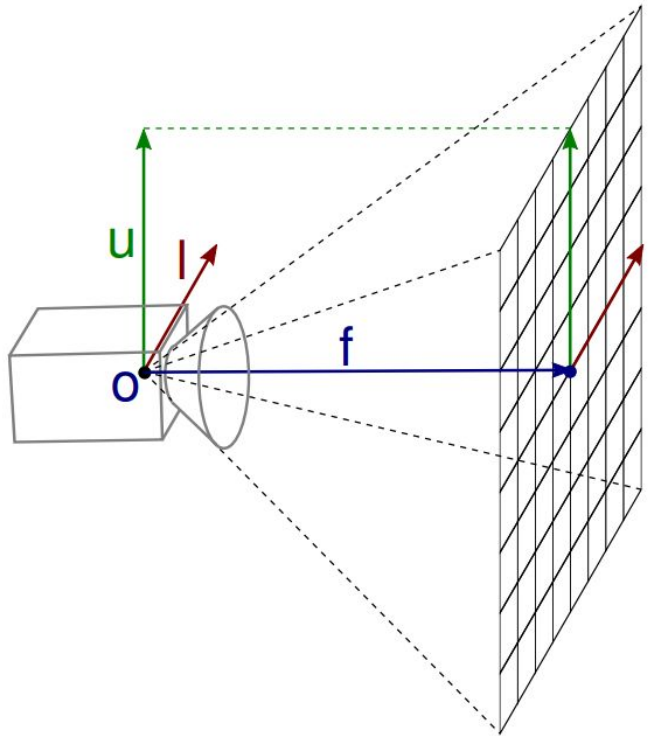
Which color do we fill each pixel with?

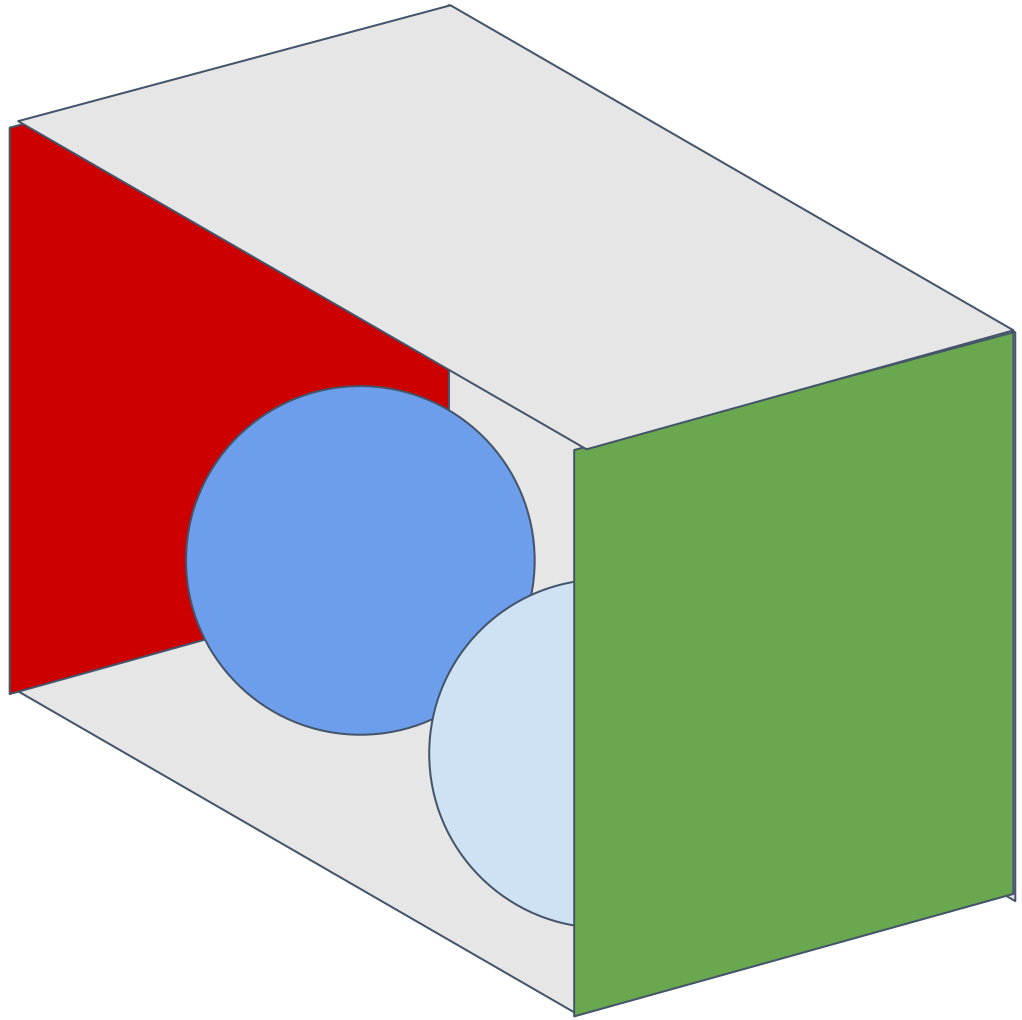
Graphics and Imaging Lab



Which color do we fill each pixel with?

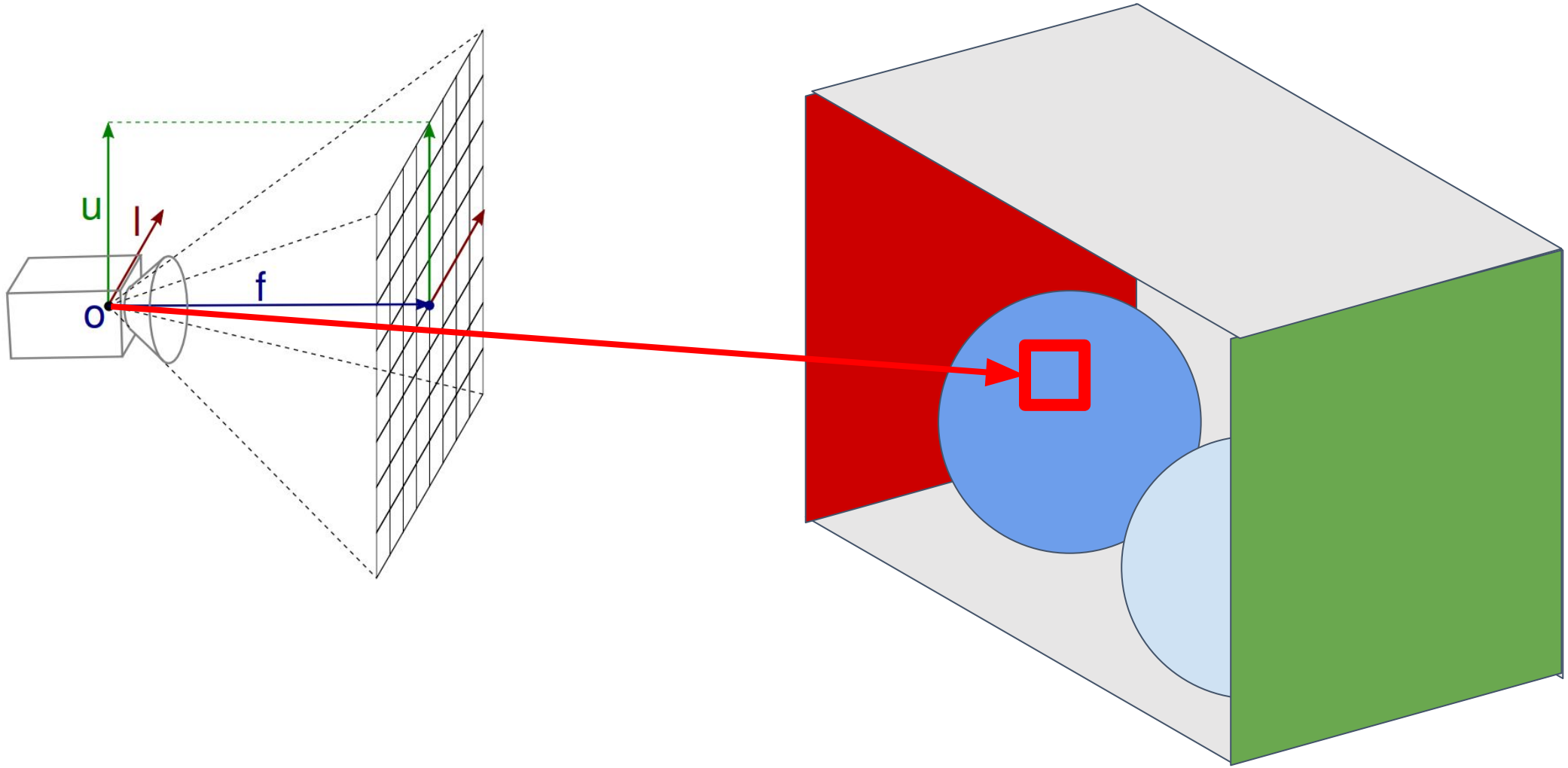
Graphics and Imaging Lab





Which color do we fill each pixel with?

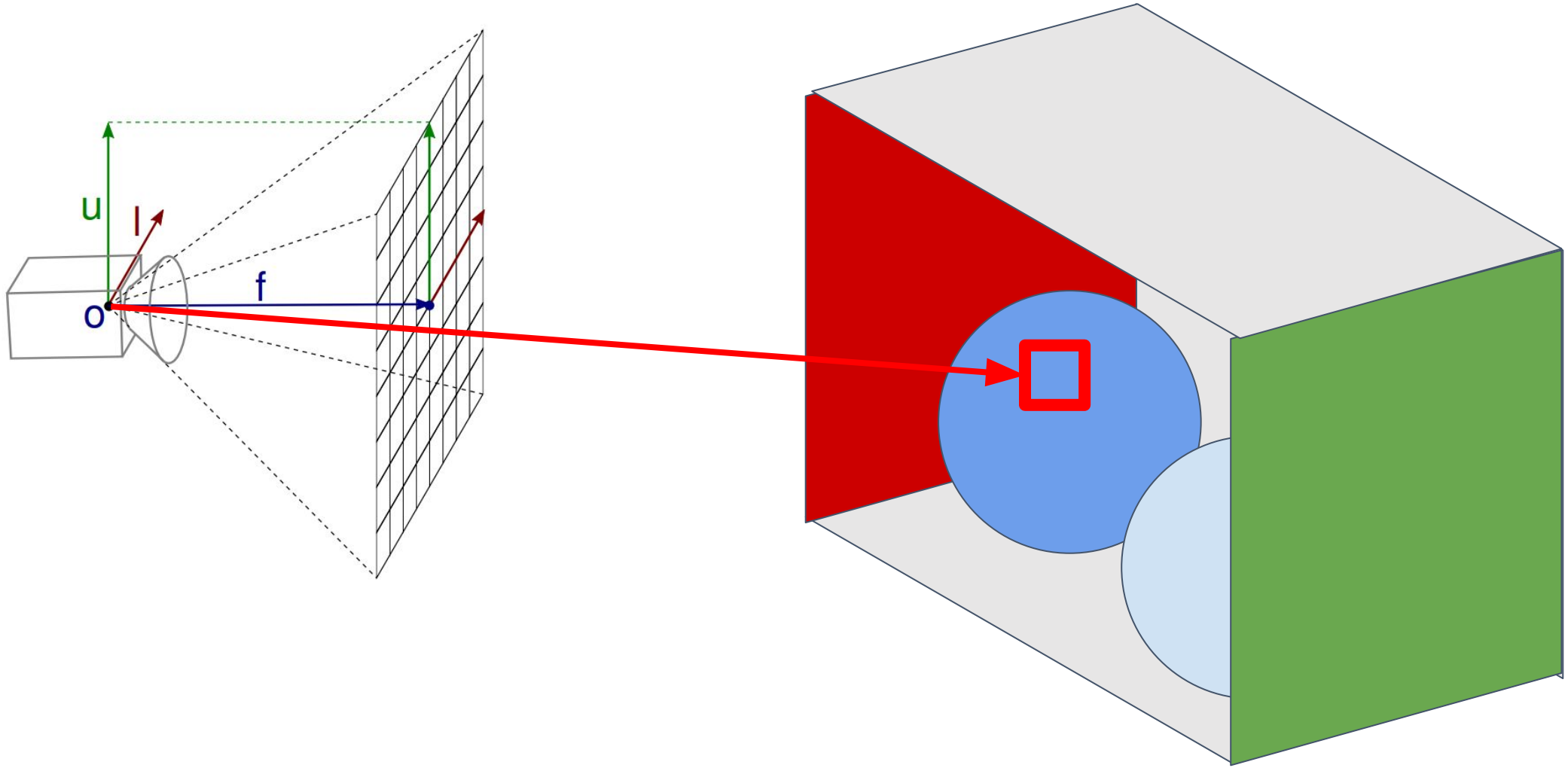
Graphics and Imaging Lab



Which color do we fill each pixel with?

Graphics and Imaging Lab

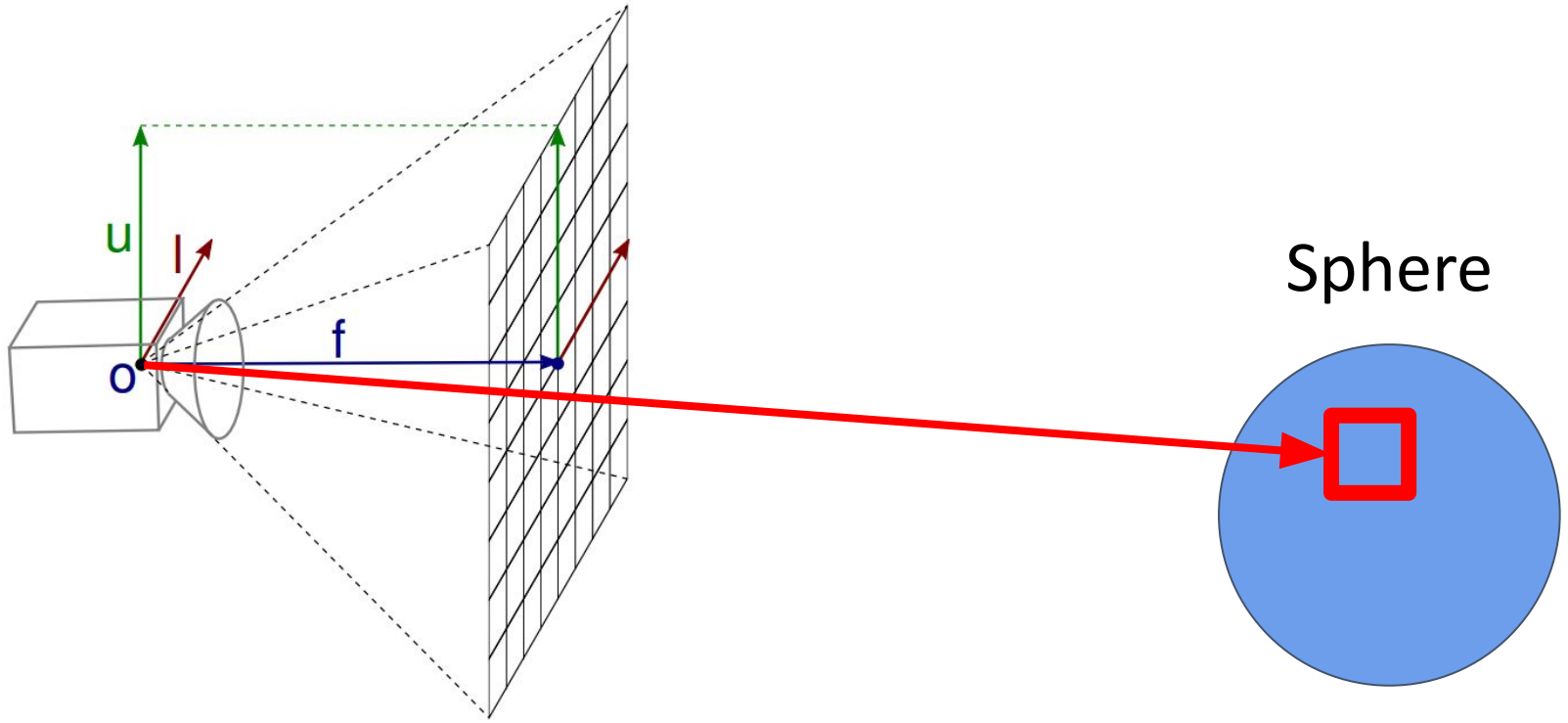
- Let's simplify



Which color do we fill each pixel with?

Graphics and Imaging Lab

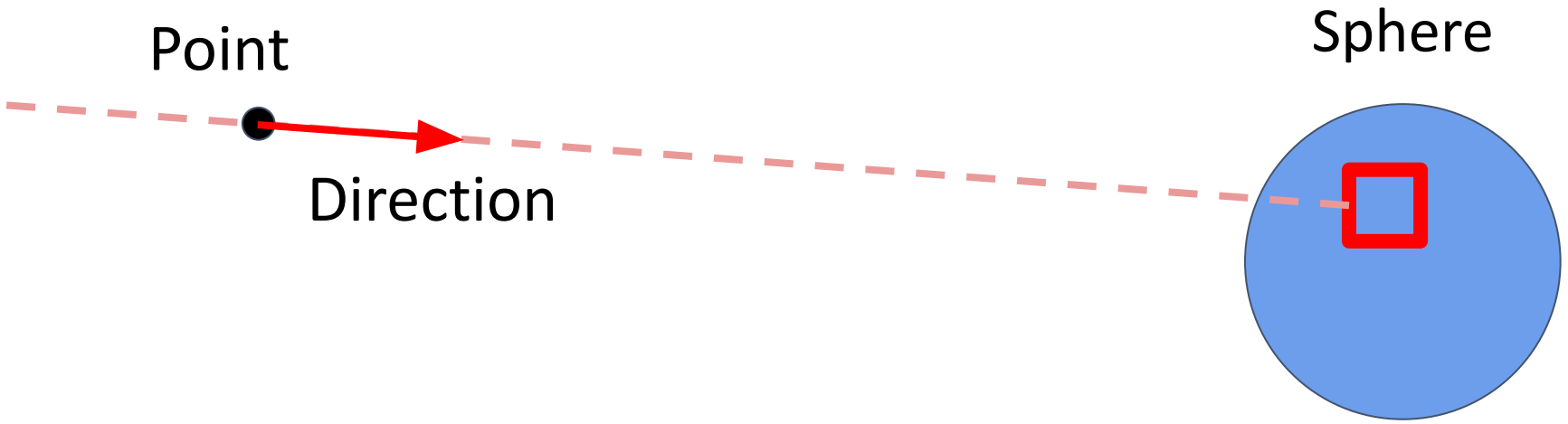
- Let's simplify



Which color do we fill each pixel with?

Graphics and Imaging Lab

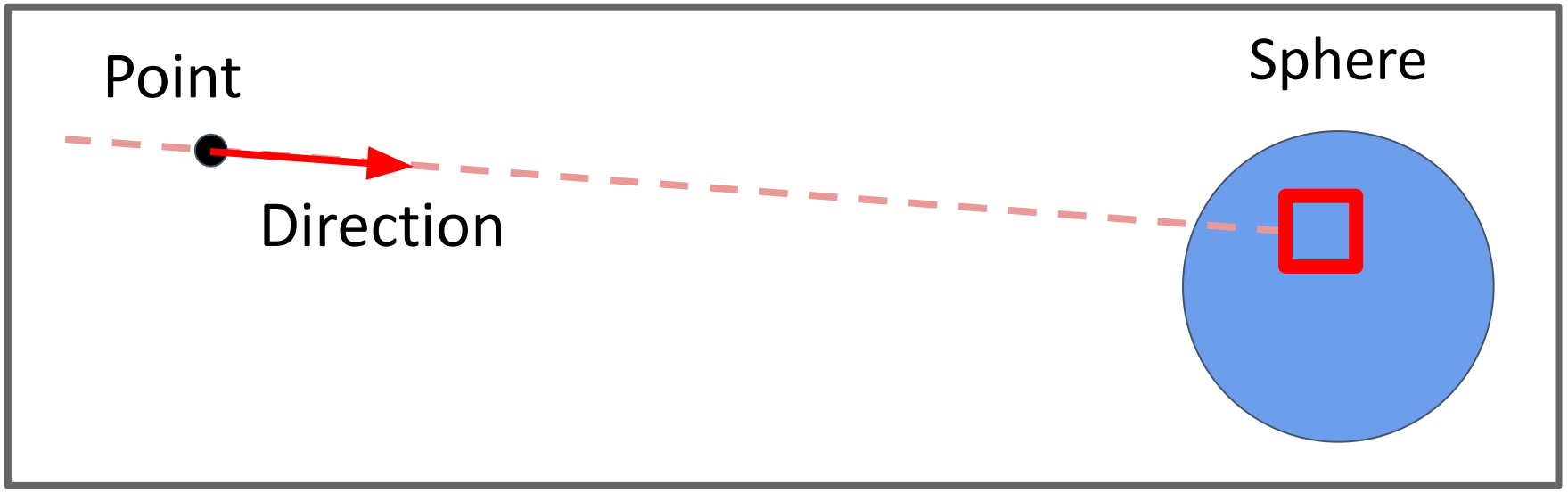
- Let’s simplify



Ray intersection problem

Graphics and Imaging Lab

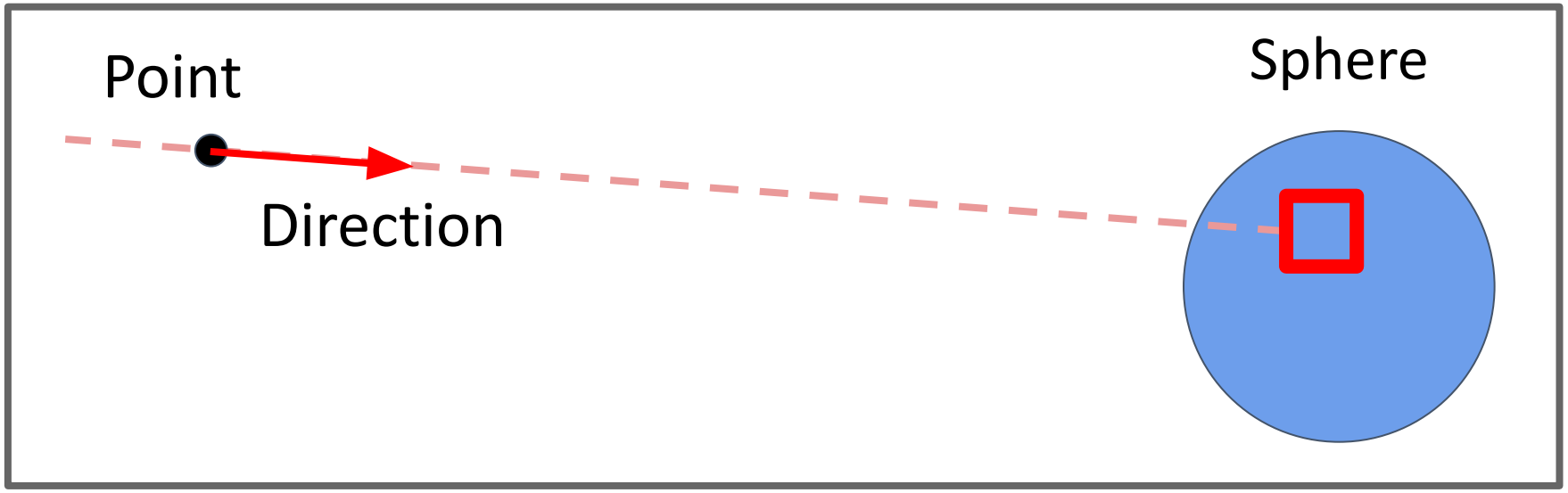
- This session we focus on ray intersections
- Inputs: ray (origin + direction), geometry (spheres or planes)



Ray intersection problem

Graphics and Imaging Lab

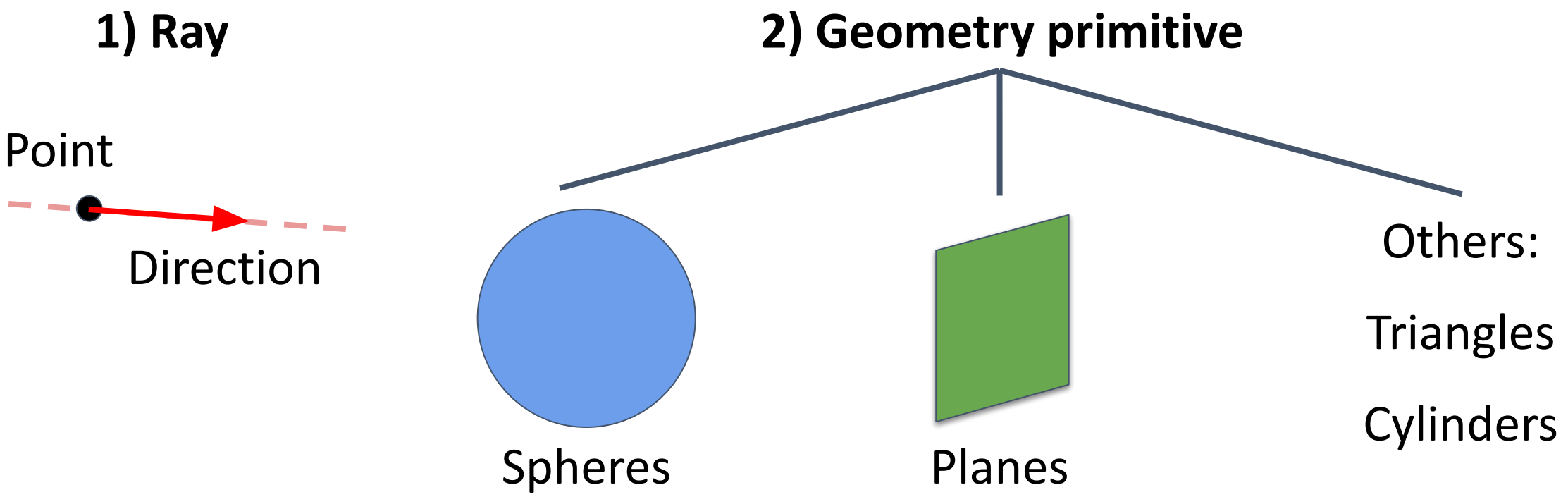
- This session we focus on ray intersections
- Inputs: ray (origin + direction), geometry (spheres or planes)



- Output: does the ray intersect? if yes, where?
  - Additional information: travel distance, surface normal, etc.

Inputs

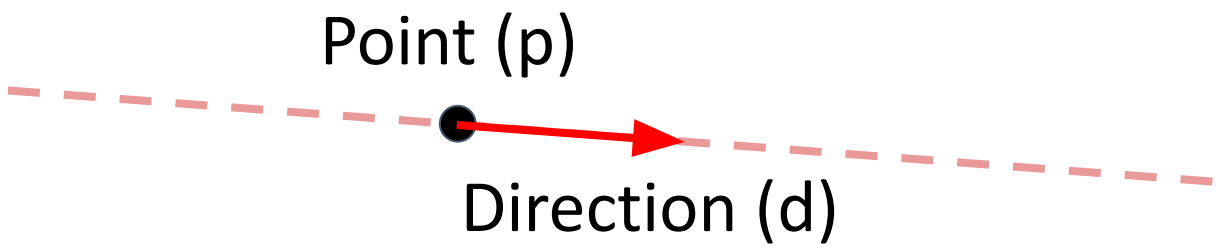
- Your intersection function should have at least two inputs:



Inputs

Graphics and Imaging Lab

- How to define a ray:



- The line - - - - is defined by
  - $p + d \cdot t$
  - $t \in \mathbb{R}$

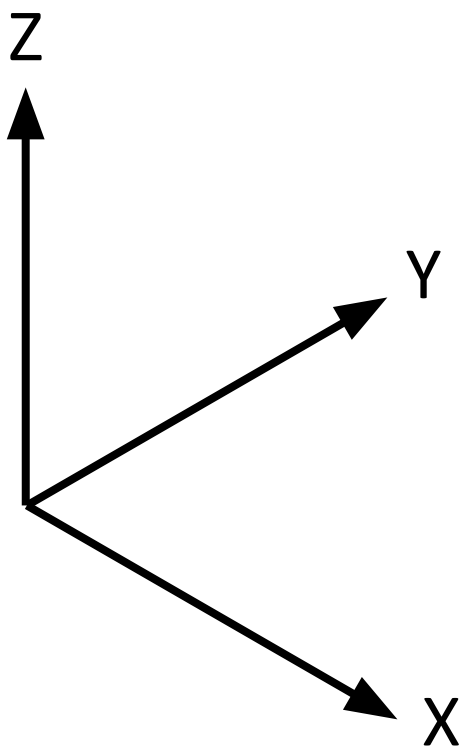
Inputs

Graphics and Imaging Lab

- How to define a geometric primitive:
  - Implicit equation

Inputs

- How to define a geometric primitive:
  - Implicit equation f(x, y, z)

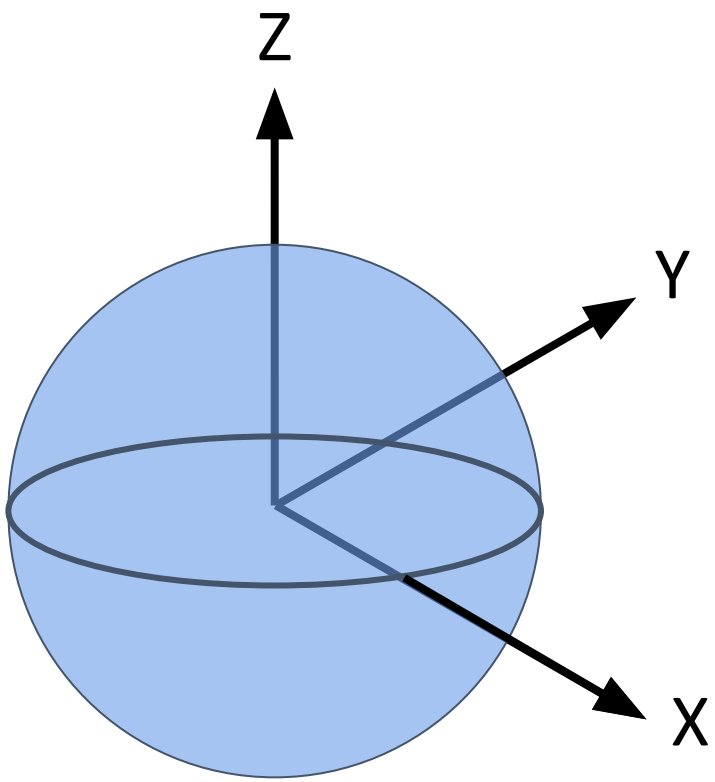


- The surface of the geometry is defined by all points (x, y, z) such that f(x, y, z) = 0

Inputs

Graphics and Imaging Lab

- How to define a geometric primitive:
  - Implicit equation f(x, y, z)

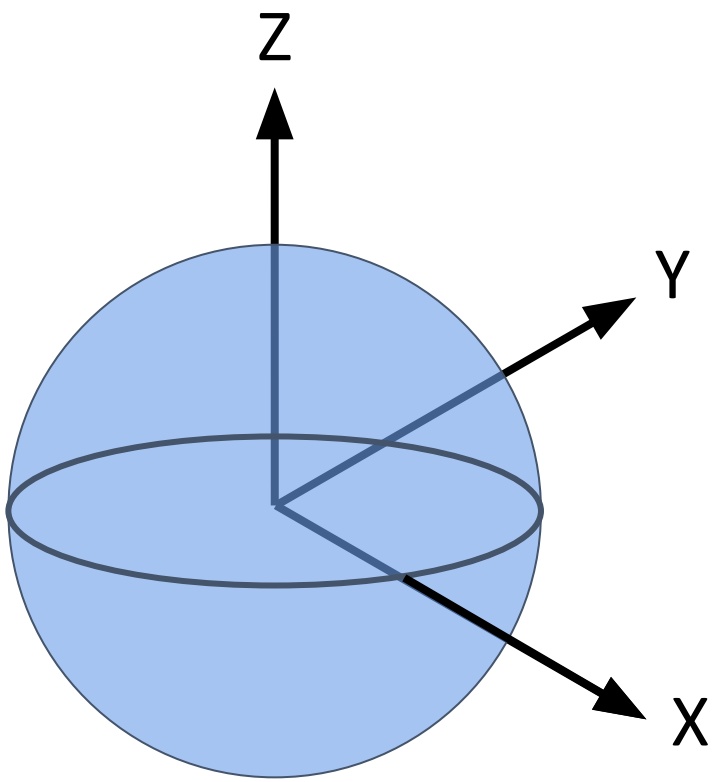


- The surface of the geometry is defined by all points (x, y, z) such that f(x, y, z) = 0
- Sphere:
  - Defined by center (c_x, c_y, c_z) and radius (r)
  - $f(x, y, z) = (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - r^2$

Inputs

Graphics and Imaging Lab

- How to define a geometric primitive:
  - Implicit equation f(x, y, z)

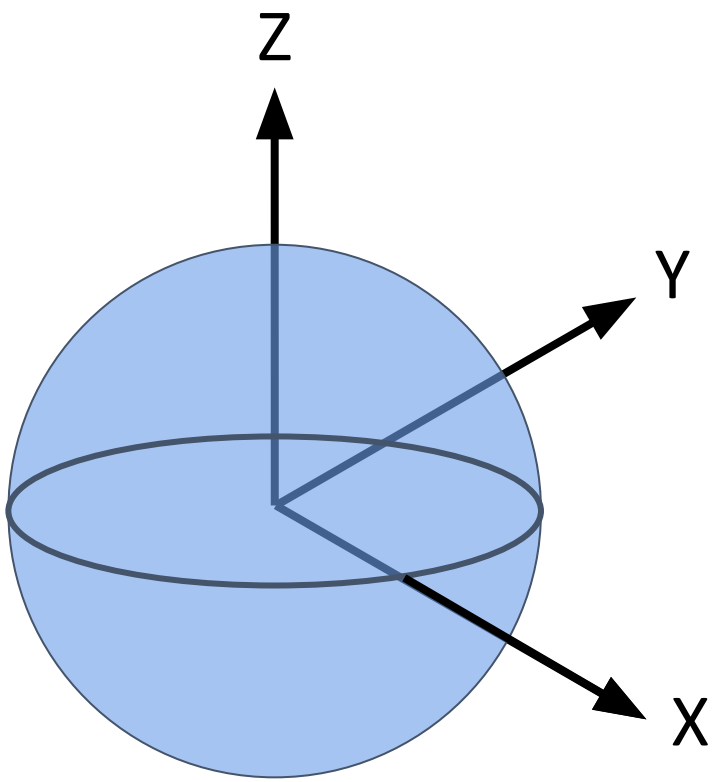


- $f(x, y, z) = (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - r^2$
- Points in the ray: $p + d \cdot t$

Inputs

Graphics and Imaging Lab

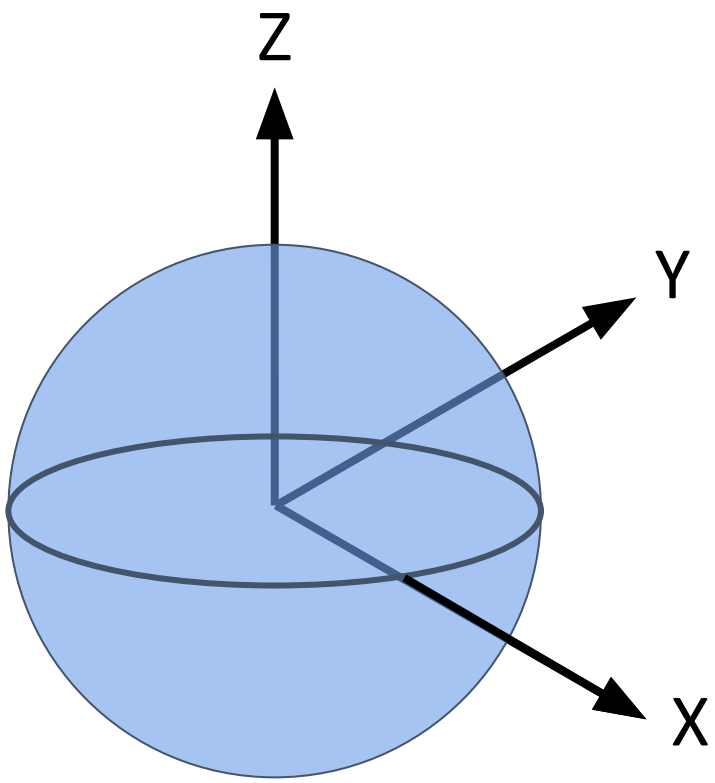
- How to define a geometric primitive:
  - Implicit equation f(x, y, z)



- $f(x, y, z) = (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - r^2$
- Points in the ray: $p + d \cdot t$
- Intersection at:
  - $f(p + d \cdot t) = 0$
  - Solve for t
  - Can have 0, 1 or 2 solutions

Inputs

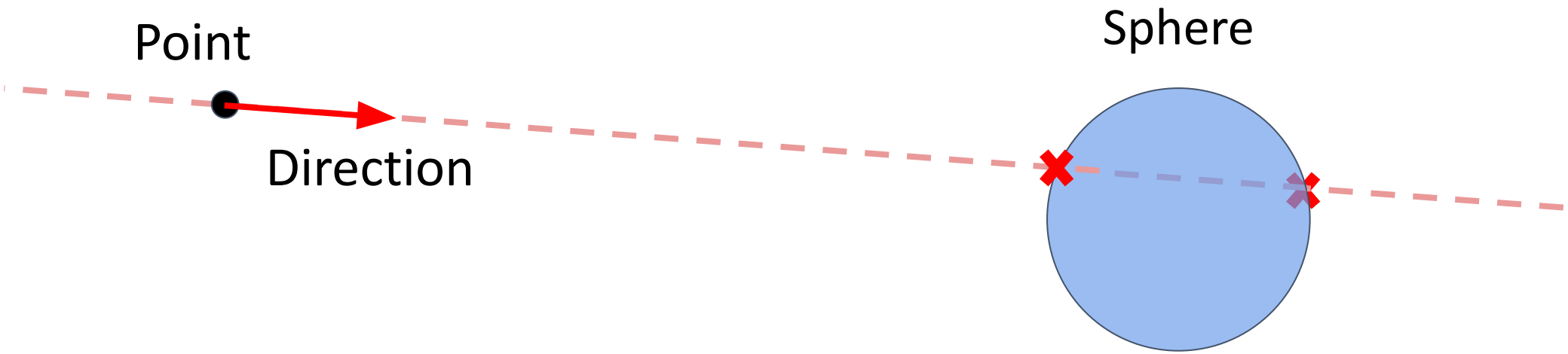
- How to define a geometric primitive:
  - Implicit equation f(x, y, z)



- $f(x, y, z) = (x - c_x)^2 + (y - c_y)^2 + (z - c_z)^2 - r^2$
- Points in the ray: $p + d \cdot t$

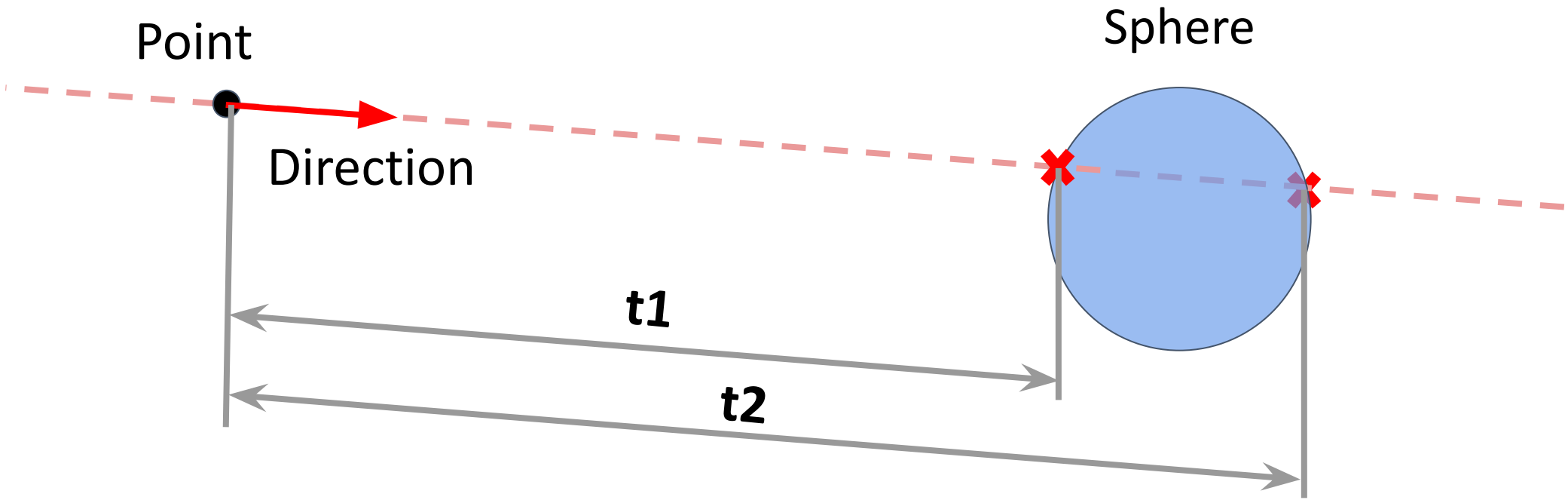
Outputs

- Solving $p + d \cdot t$ gives the distance to the sphere



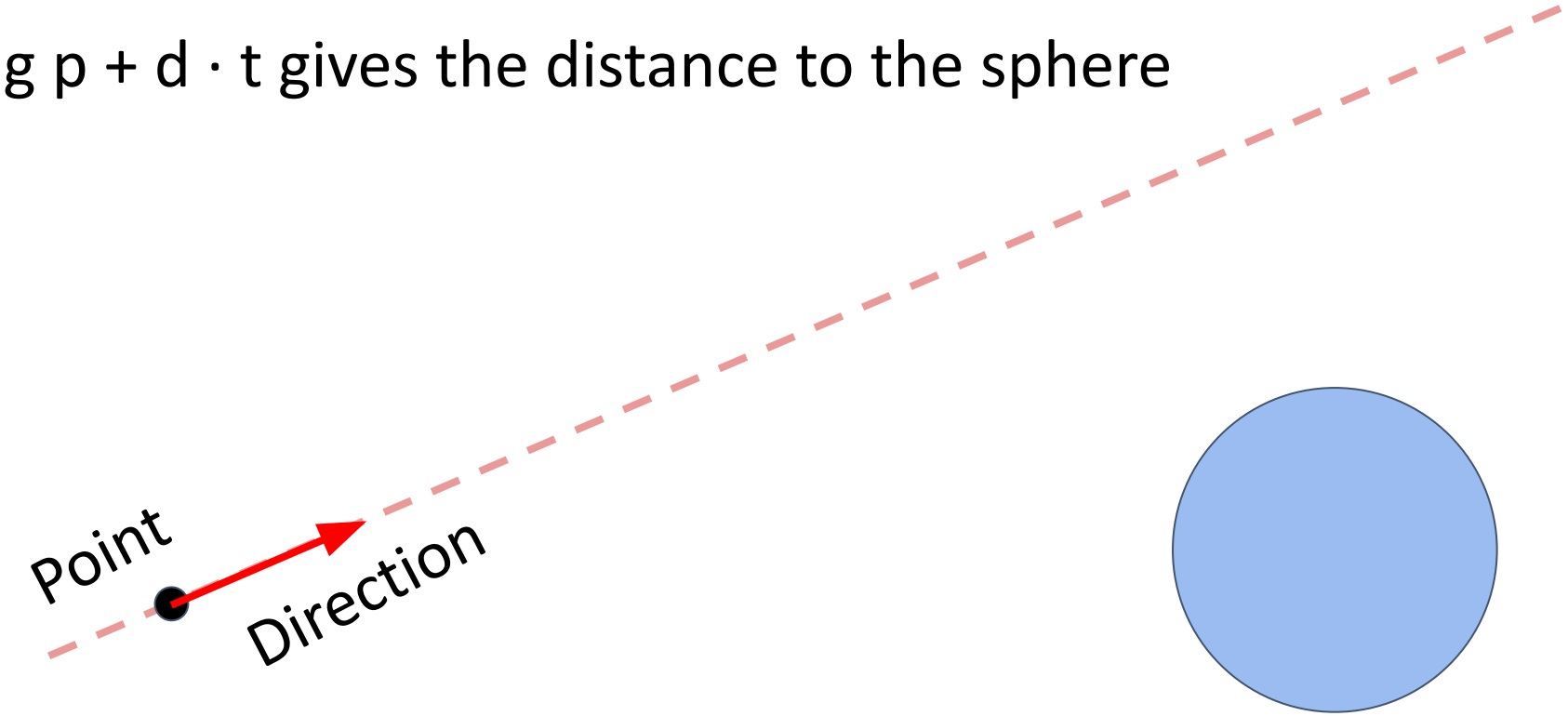
Outputs

- Solving p + d · t gives the distance to the sphere



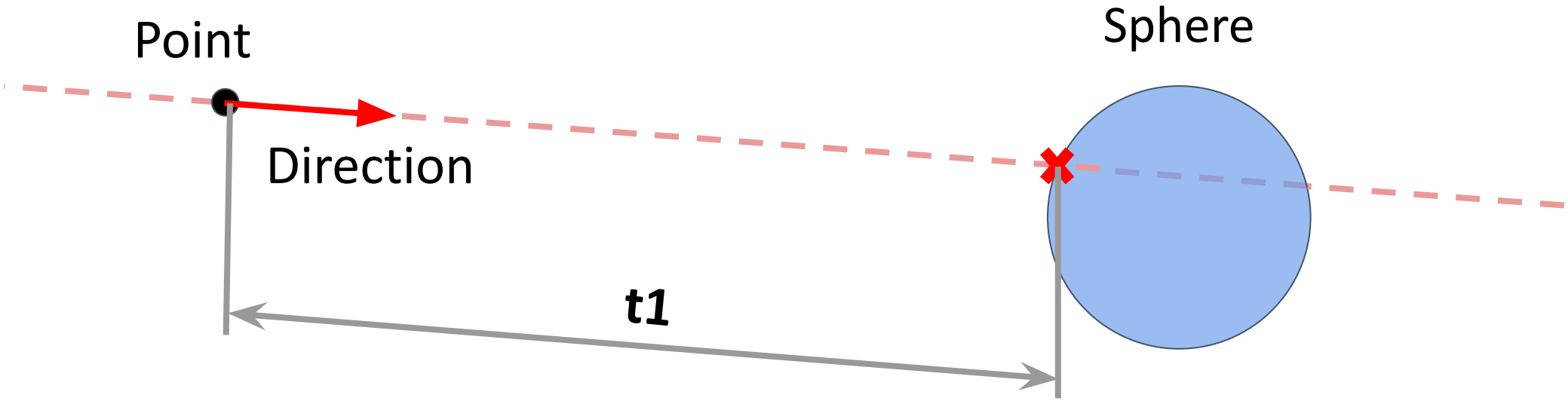
Outputs

- Solving p + d · t gives the distance to the sphere



Outputs

- Solving p + d · t gives the distance to the sphere



- Distance: t1
- Intersection point: p + d · t1
- Surface normal at t1, etc.

Questions

Graphics and Imaging Lab

DO ASK questions, either now or after the lab

But be reasonable, please :)

pluesia@unizar.es | dsubias@unizar.es | o.pueyo@unizar.es

What to expect from this session

Graphics and Imaging Lab

In the programming language of your choice, implement:

- **Primitives:** spheres and planes
- **Ray:** point + direction + other stuff you may want
- One **intersect function** per type of primitive
  - Inputs: Ray, one primitive
  - Output: Does the ray intersect? If yes, where? + additional info
- Recommended deadline: **October 18th**.
  - **Next lab:** define a camera and generate an image
  - **Extensions** (do not count towards deadline):
    - Other primitives: cones, cylinders, ellipsoids, disks or triangles
    - Acceleration structures: bounding volumes, multi-threading, etc.
    - Constructive solid geometry: google it or ask us :)