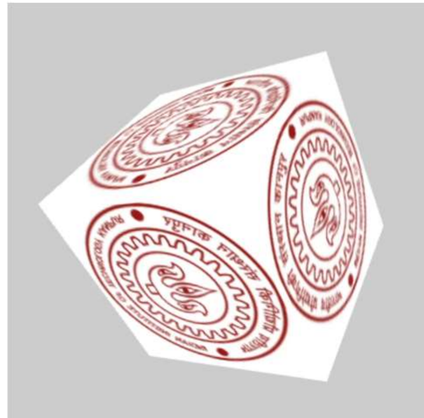


# Introduction to Computer Graphics (CS360A)


Instructor: Soumya Dutta

Department of Computer Science and Engineering

Indian Institute of Technology Kanpur (IITK)

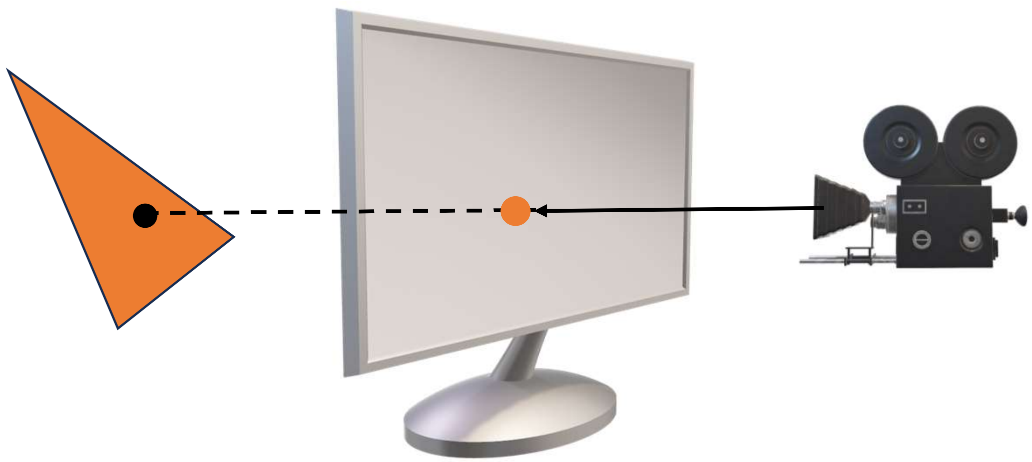
email: soumyad@cse.iitk.ac.in

# Acknowledgements

- A subset of the slides that I will present throughout the course are adapted/inspired by excellent courses on Computer Graphics offered by Prof. Han-Wei Shen, Prof. Wojciech Matusik, Prof. Frédo Durand, Prof. Abe Davis, and Prof. Cem Yuksel

# Ray Tracing

- Start a ray from camera and find out which scene object the ray hits
- If a hit is found, set the fragment color with the primitive color at that point
- If no hit detected, set the fragment with background color

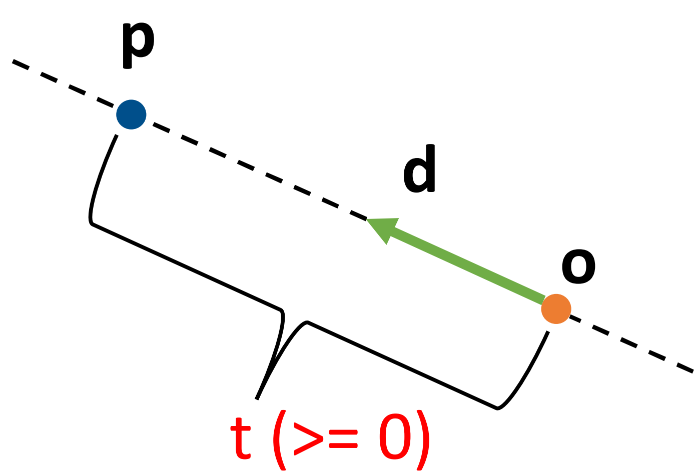


# What is a Ray?

- A ray is simply a point with a direction
- Using a parametric representation, we can walk along the ray

$$\mathbf{p} = \mathbf{o} + t\mathbf{d}$$

- t is the distance between origin **o** and point p along the ray if **d** is unit vector
- t < 0 means, opposite to the direction **d**



# Ray Intersections

IITK CS360A

- In ray tracing, we need to compute the intersection of the ray with the objects in the scene
  - Objects can be made of triangles
- Objects in the scene can also be represented by implicit functions
  - Spheres, Planes, etc.
- Complicated objects are still represented as triangular mesh

# Ray Intersections

IITK CS360A

- Implicit functions: $f(\mathbf{p}) = 0$
- Point on a ray: $\mathbf{p} = \mathbf{o} + t\mathbf{d}$

$$f(\mathbf{o} + t\mathbf{d}) = 0$$

- If there is a t for which the above equation is satisfied, we can say that the ray intersects with the implicit surface, i.e., we have found a **hit**
- Else, if no such t exists, then it is a **miss**

# Ray Sphere Intersection

$$f(\mathbf{p}) = 0$$

# Ray Sphere Intersection

$$f(\mathbf{p}) = x^2 + y^2 + z^2 - r^2 = 0$$

# Ray Sphere Intersection

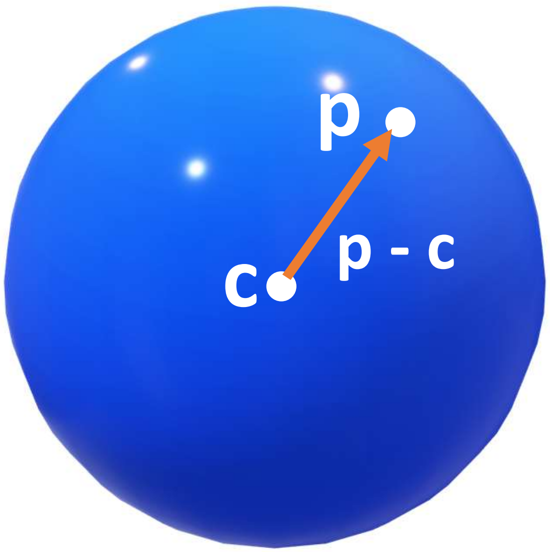
$f(\mathbf{p}) = \mathbf{p} \, . \, \mathbf{p} - r^2 = 0$ Sphere at the origin (0,0,0)

# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$

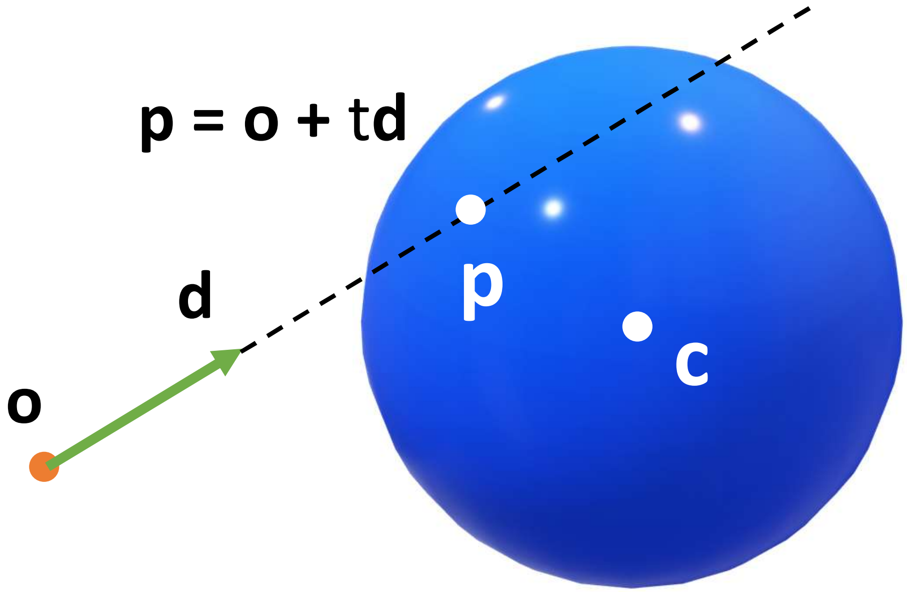
# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$



# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$



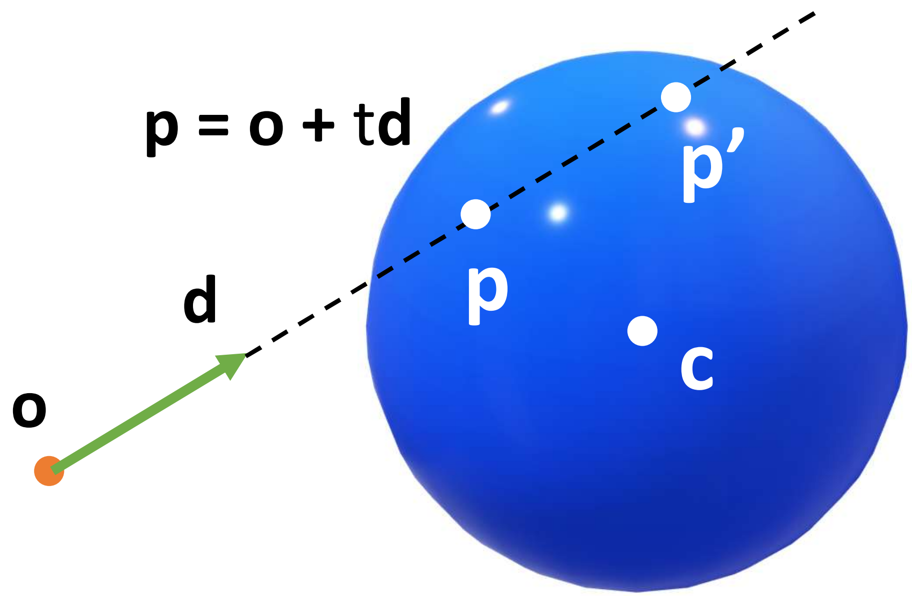

$$(\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$

$$(\mathbf{o} + t\mathbf{d} - \boldsymbol{c}).(\mathbf{o} + t\mathbf{d} - \boldsymbol{c}) - r^2 = 0$$

$$(\mathbf{d}\,.\,\mathbf{d})t^2 + 2\mathbf{d}(\mathbf{o} - \boldsymbol{c})t + (\mathbf{o} - \boldsymbol{c}).(\mathbf{o} - \boldsymbol{c}) - r^2 = 0$$

# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$

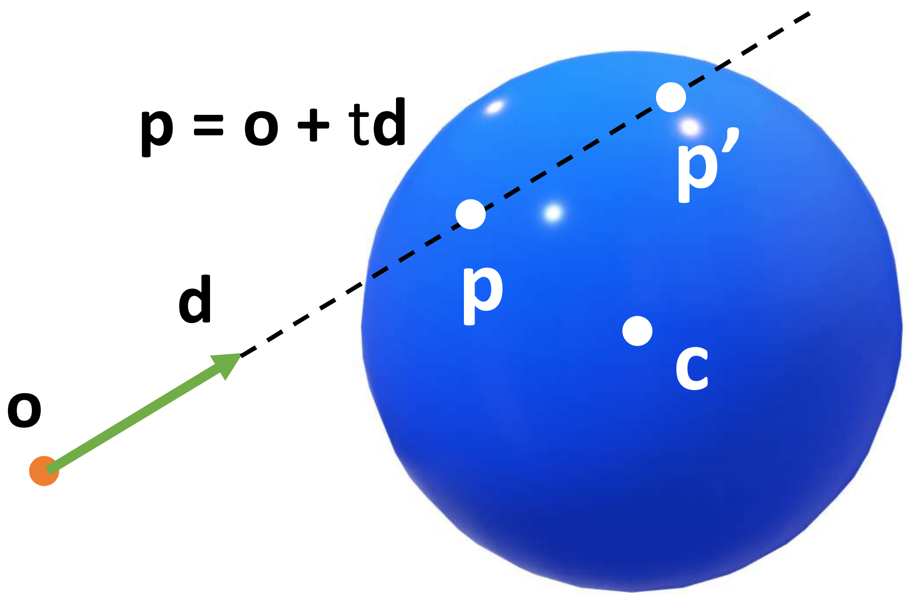



$$(\mathbf{d}\,.\,\mathbf{d})t^2 + 2\mathbf{d}(\mathbf{o} - \boldsymbol{c})t + (\mathbf{o} - \boldsymbol{c}).(\mathbf{o} - \boldsymbol{c}) - r^2 = 0$$

- This is a quadratic equation of t
- It can have at most two solutions
- Two solutions make sense since the ray can hit the sphere twice

# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$




$$(\mathbf{d}\,.\,\mathbf{d})t^2 + 2\mathbf{d}(\mathbf{o} - \boldsymbol{c})t + (\mathbf{o} - \boldsymbol{c}).(\mathbf{o} - \boldsymbol{c}) - r^2 = 0$$

$$\color{red}{a}t^2 + \color{red}{b}t + \color{red}{c} = 0$$

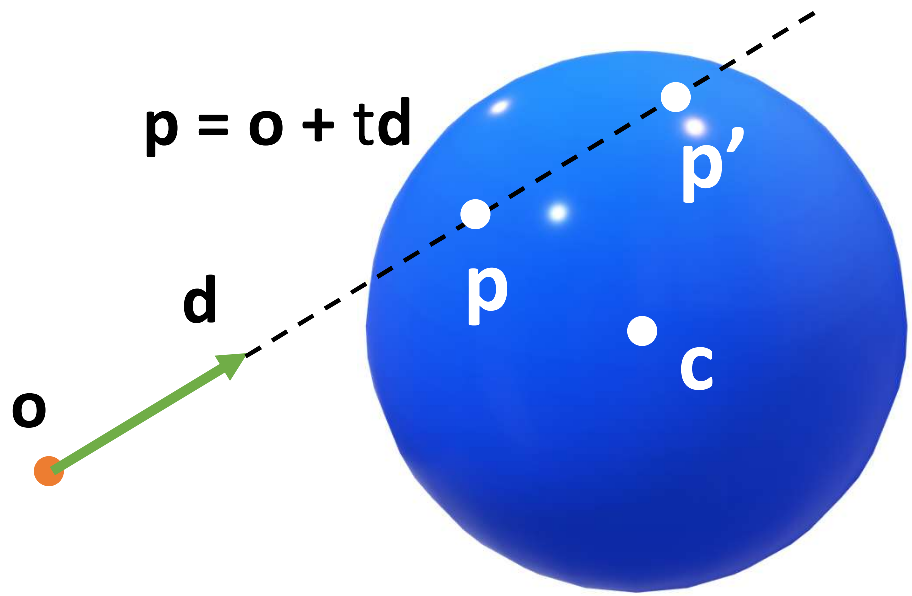
$$\color{red}{a} = (\mathbf{d}\,.\,\mathbf{d})$$

$$\color{red}{b} = 2\mathbf{d}(\mathbf{o} - \boldsymbol{c})$$

$$\color{red}{c} = (\mathbf{o} - \boldsymbol{c}).(\mathbf{o} - \boldsymbol{c}) - r^2$$

# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$




$$(\mathbf{d}\,.\,\mathbf{d})t^2 + 2\mathbf{d}(\mathbf{o} - \boldsymbol{c})t + (\mathbf{o} - \boldsymbol{c}).(\mathbf{o} - \boldsymbol{c}) - r^2 = 0$$

$$at^2 + bt + c = 0$$

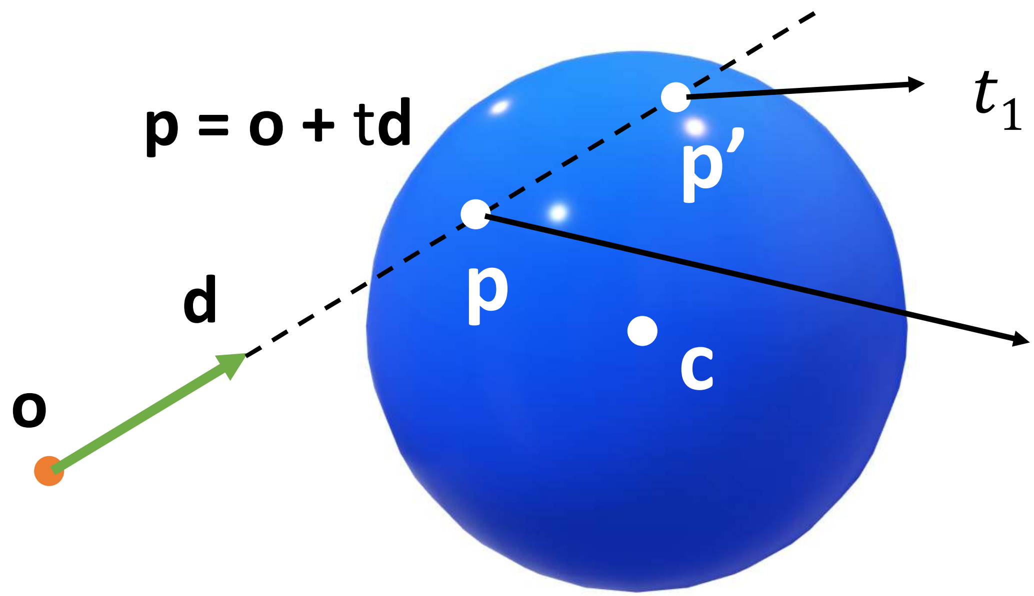
$$t = \frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$$

When $b^2 - 4ac$ >= 0, we have hit

If $b^2 - 4ac$ < 0, the ray misses the sphere

# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$

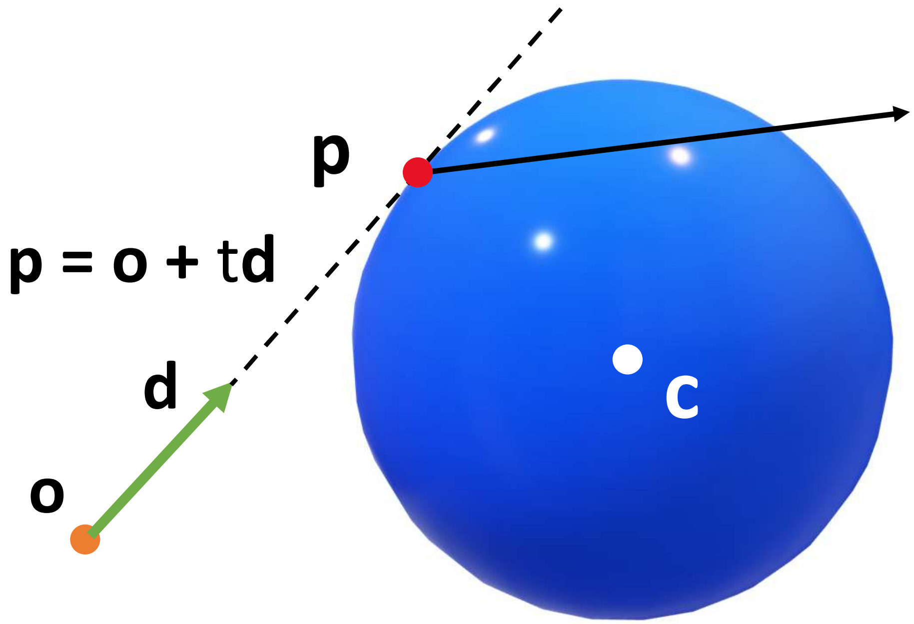



$$t_1 = \frac{-b + \sqrt{b^2 - 4ac}}{2a}$$

$$t_2 = \frac{-b - \sqrt{b^2 - 4ac}}{2a}$$

# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$



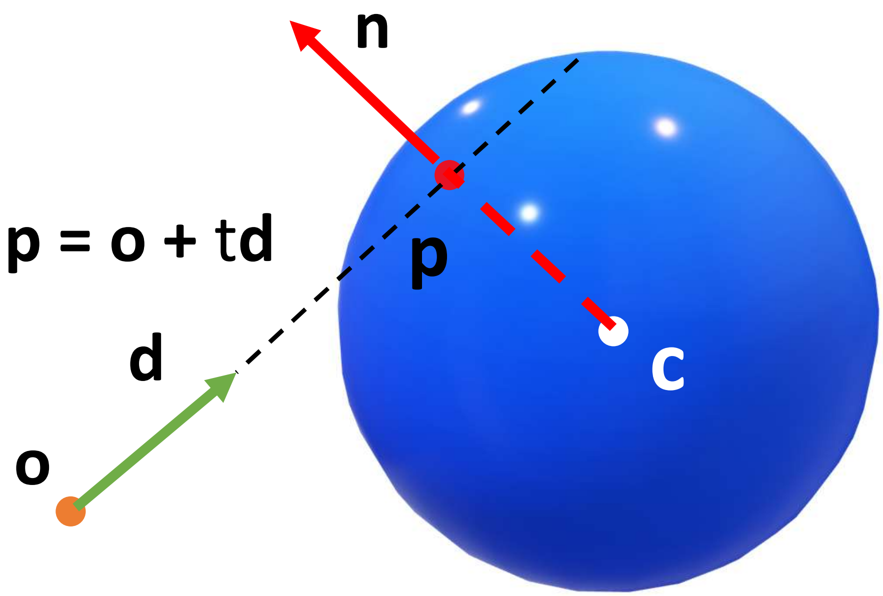

$$t = \frac{-b}{2a}$$

When $b^2 - 4ac$ = 0, we have one solution

The ray is tangent to the sphere surface

# Ray Sphere Intersection

$$f(\mathbf{p}) = (\mathbf{p} - \boldsymbol{c}).(\mathbf{p} - \boldsymbol{c}) - r^2 = 0$$



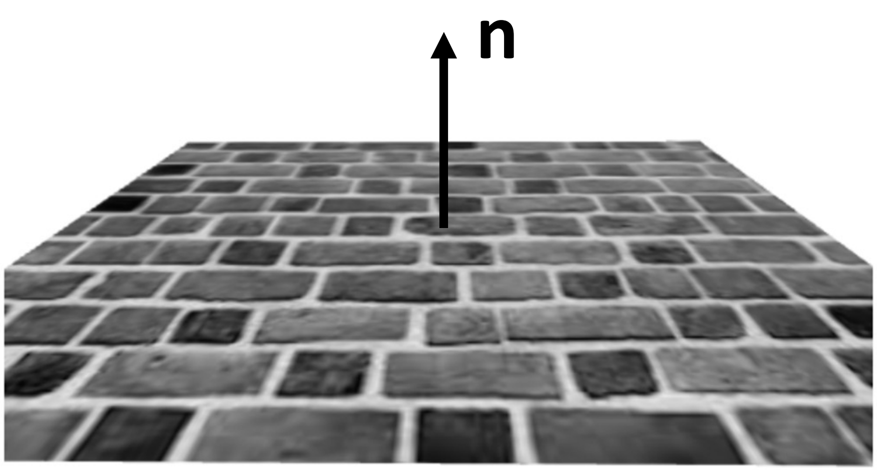

Compute the normal at the intersection point for shading/illumination

**n =** normalize**(p – c)**

# Ray Plane Intersection

$$f(\mathbf{p}) = 0$$

$$\mathbf{p} \,.\, \mathbf{n} - \mathrm{c} = 0$$

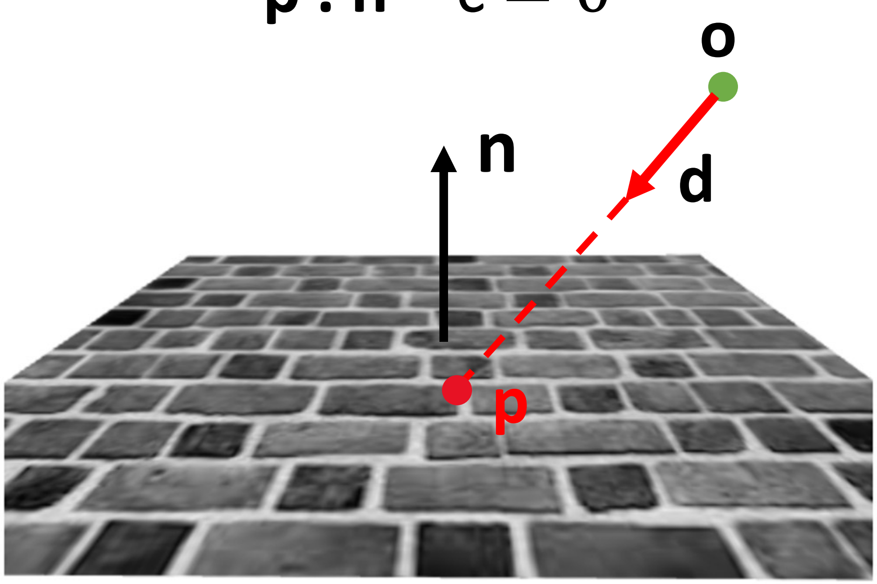



- c is is the perpendicular distance of the plane from the origin
- **p** is a position vector lying on the plane

# Ray Plane Intersection

$f(\mathbf{p}) = 0$

$\mathbf{p} \, . \, \mathbf{n} - \mathrm{c} = 0$




- c is is the perpendicular distance of the plane from the origin
- **p** is a position vector lying on the plane

$$\mathbf{p} = \mathbf{o} + \mathrm{t}\mathbf{d}$$

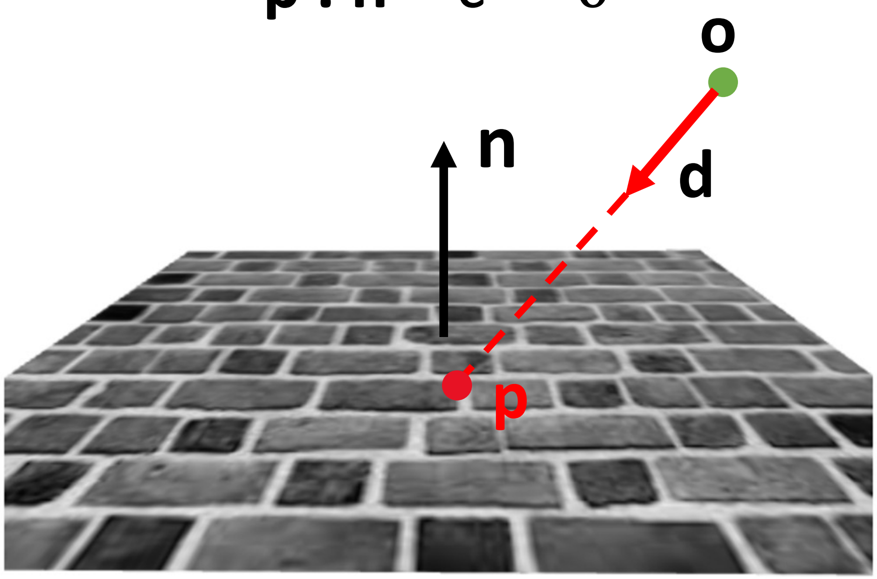
$$(\mathbf{o} + \mathrm{t}\mathbf{d}) \, . \, \mathbf{n} - \mathrm{c} = 0$$

A linear equation in t

# Ray Plane Intersection

$$f(\mathbf{p}) = 0$$

$$\mathbf{p} \,.\, \mathbf{n} - \mathrm{c} = 0$$




- c is is the perpendicular distance of the plane from the origin
- **p** is a position vector lying on the plane

$$\mathbf{p} = \mathbf{o} + \mathrm{t}\mathbf{d}$$

$$(\mathbf{o} + \mathrm{t}\mathbf{d}) \,.\, \mathbf{n} - \mathrm{c} = 0$$

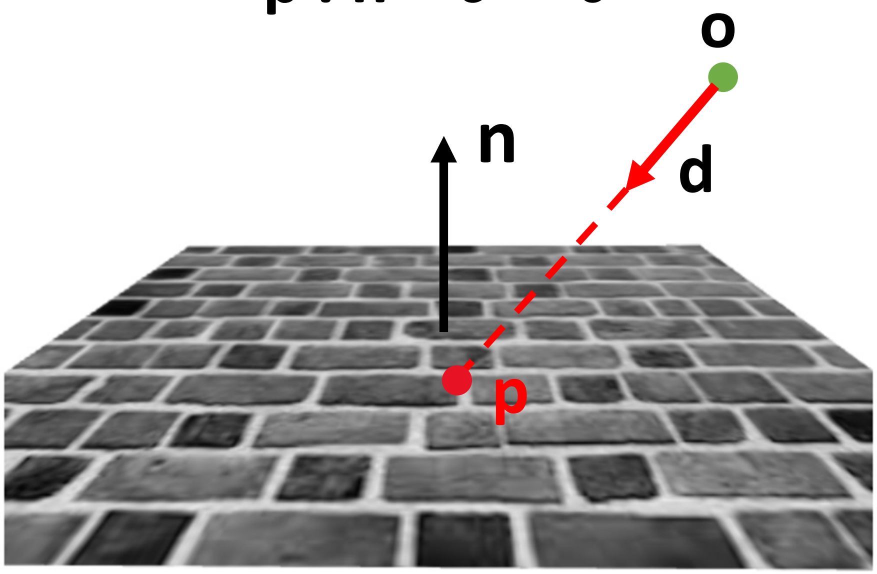
$$t = \frac{c - \boldsymbol{o}.\boldsymbol{n}}{\boldsymbol{d}.\boldsymbol{n}}$$

Do we always have a solution?

# Ray Plane Intersection

$$f(\mathbf{p}) = 0$$

$$\mathbf{p}\,.\,\mathbf{n} - \mathrm{c} = 0$$



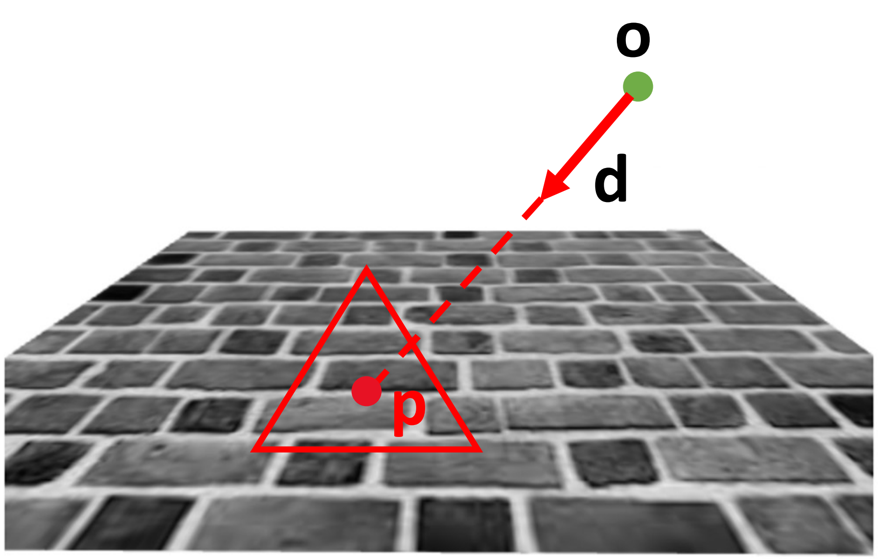

- If the plane is axis aligned, say with x-y plane
- Then, **n** vector will be [0,0,1]

$$t = \frac{c - \boldsymbol{o}.\boldsymbol{n}}{\boldsymbol{d}.\boldsymbol{n}}$$

$$t = \frac{c - oz}{d_z}$$

# Ray Triangle Intersection

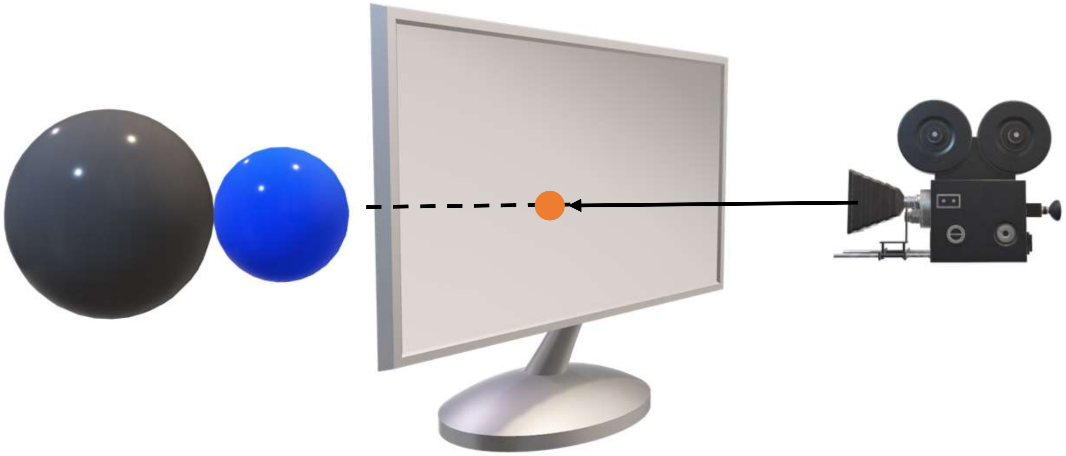
- Many ways to do this, let us see one of such method




- Using triangle coordinates, we can form a plane
- Use cross product of two planer vectors along two edges of triangle to compute the normal vector
- Solve for Ray Plane intersection
- Use Barycentric Coordinate of the solution to ensure of the point is inside the triangle or not

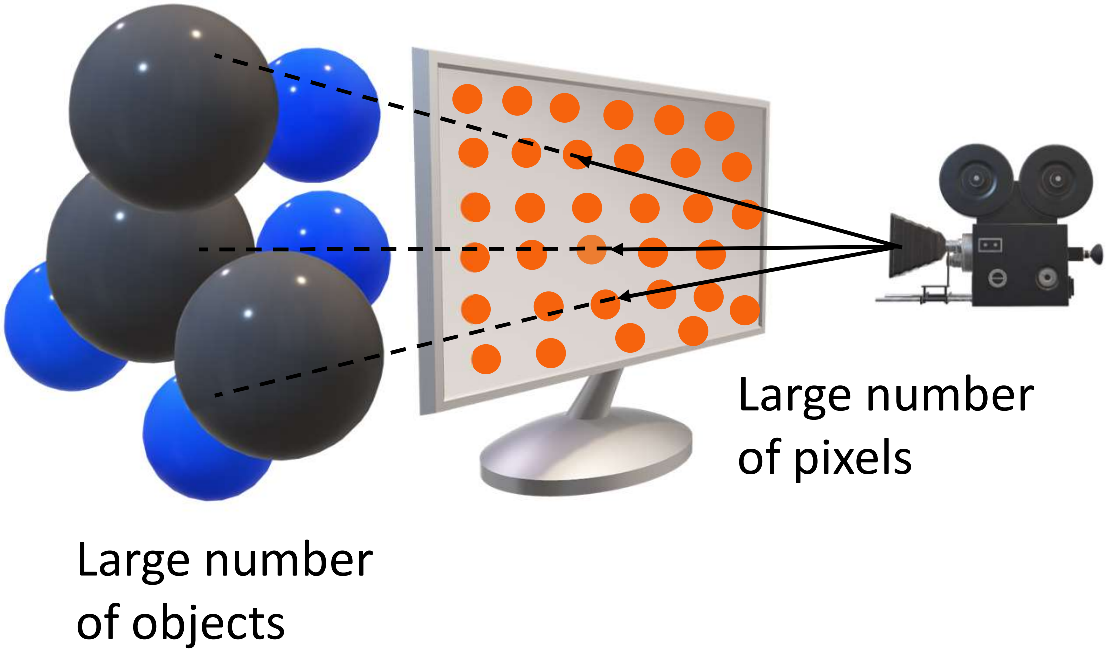
# Ray Tracing Algorithm

- For each ray
  - For each object
    - If ray hits any object
      - Find the **closest** object
  - Return min hit distance



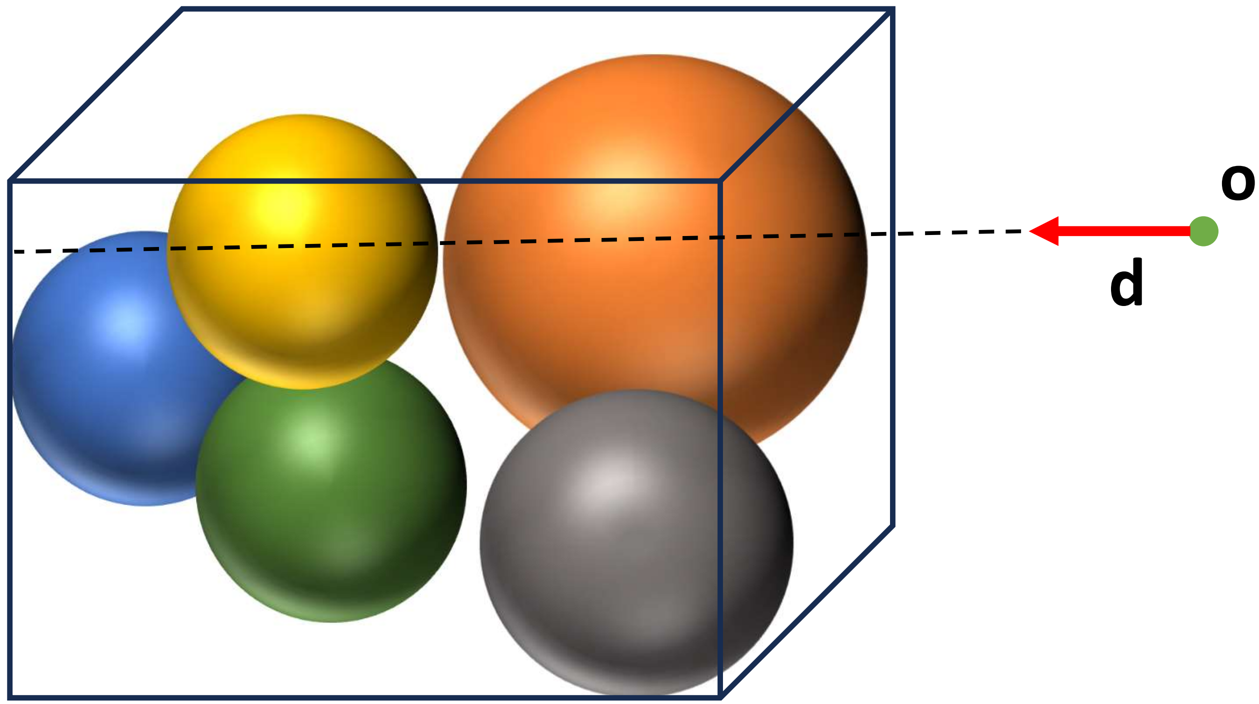
# Ray Tracing Algorithm

- There can be millions of objects in our scene
- There can be millions of pixels in our output image
- Brute force ray tracing will be extremely slow, and this is why historically it has been treated as a post-processing technique and not suitable for real-time graphics



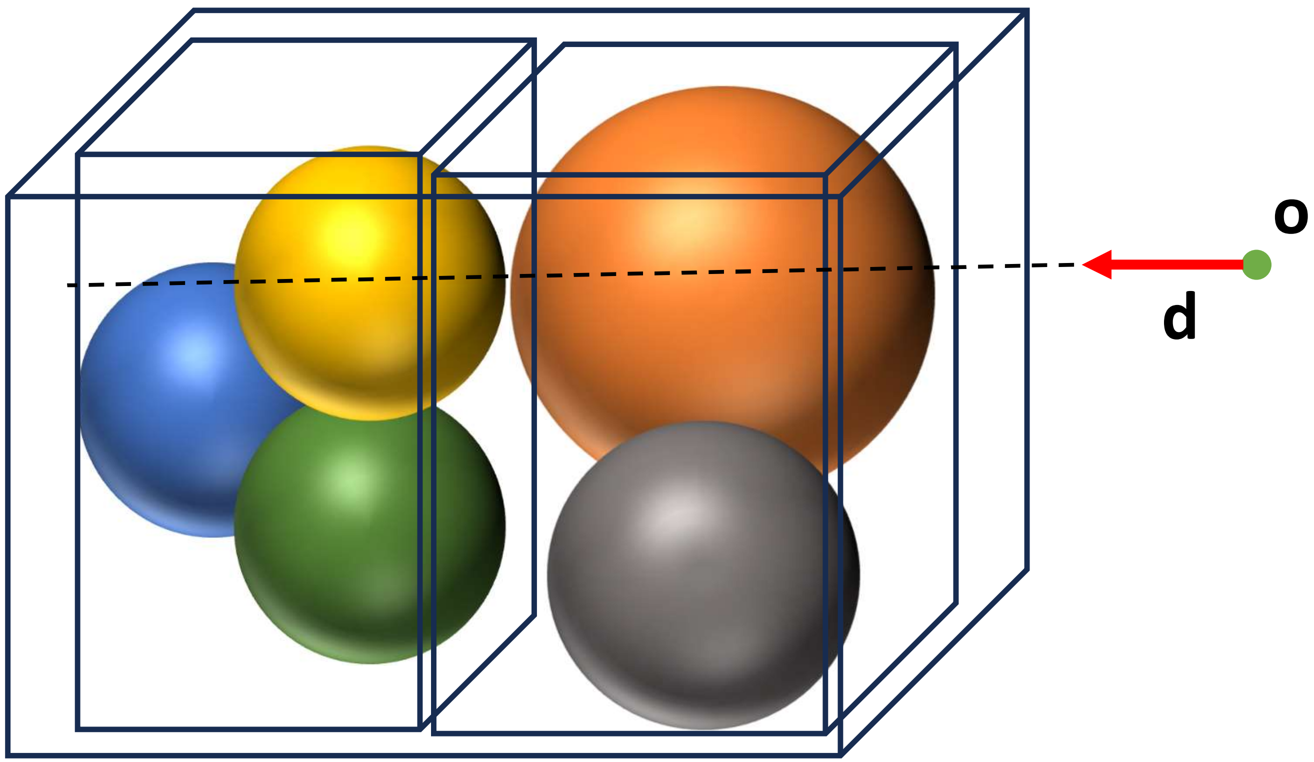
# Ray Tracing Accelerations

- Space partitioning and acceleration structures
- Axis Aligned Bounding Box (AABB)

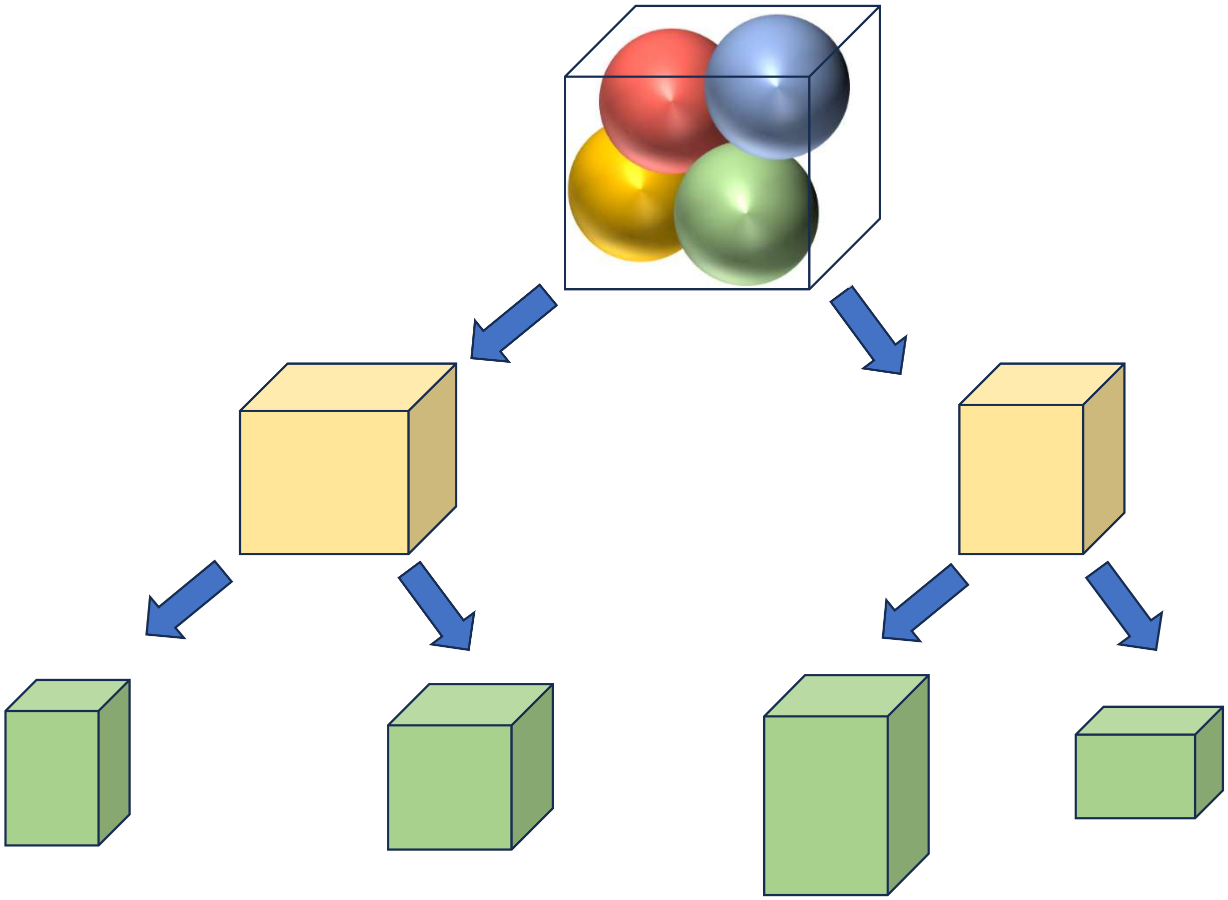


# Ray Tracing Accelerations

- Space partitioning and acceleration structures
- Axis Aligned Bounding Box (AABB)

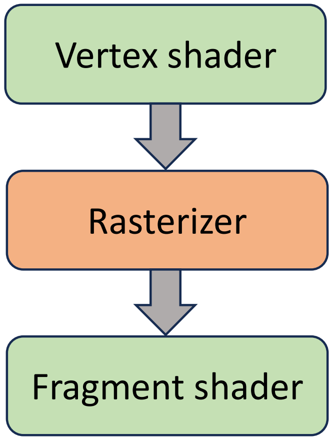


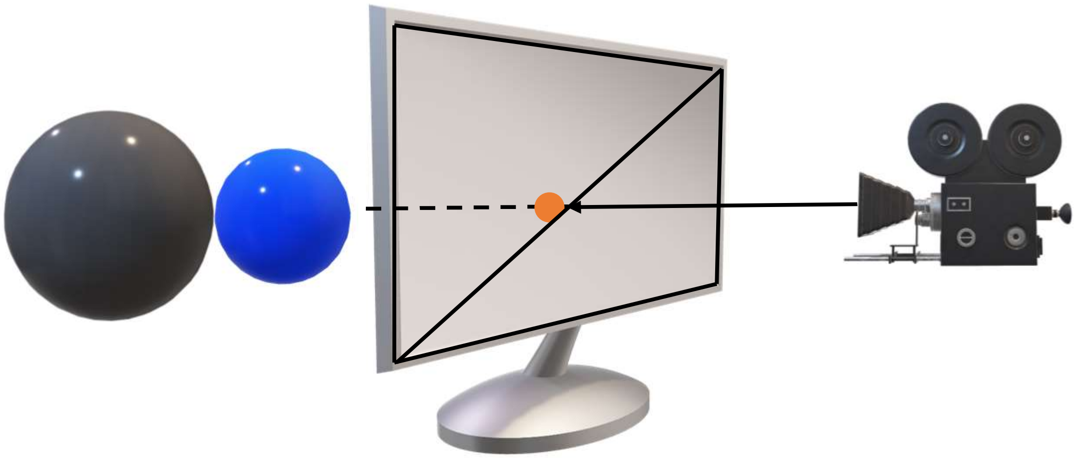
# Bounding Volume Hierarchy (BVH)



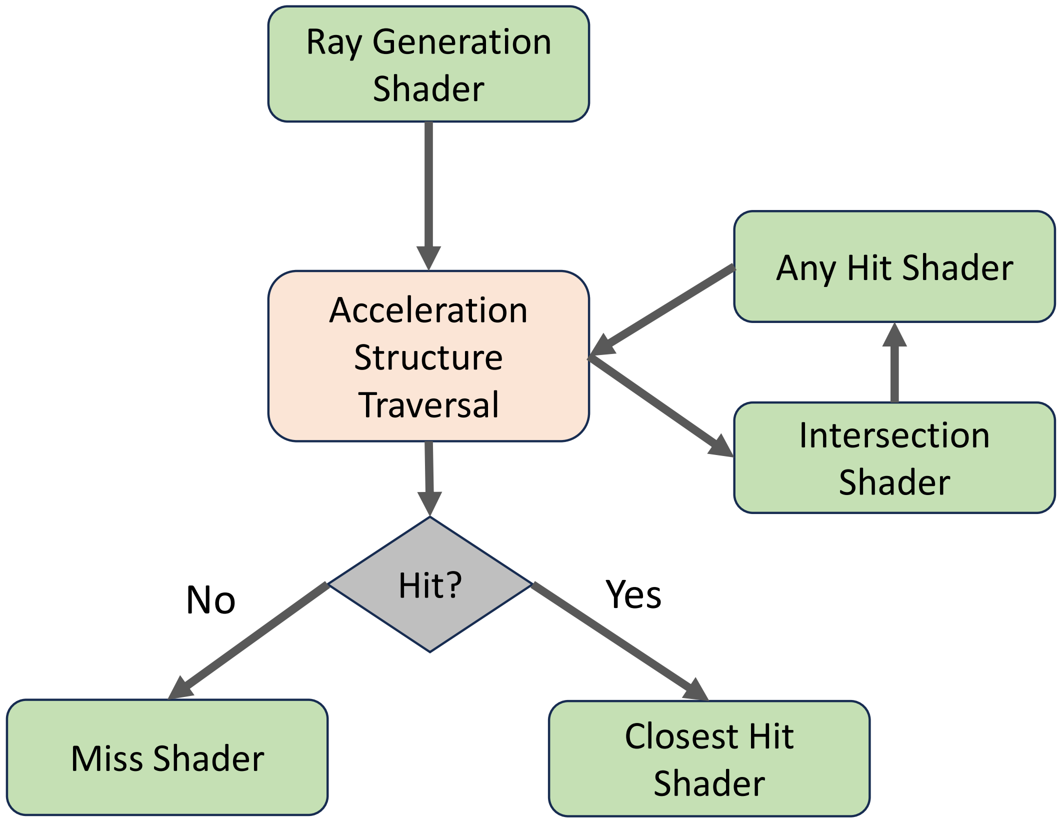
# (Software) Ray Tracing in GLSL

- Render a quad on the entire screen so that we have a fragment shader running for each fragment
- Implement ray tracing in fragment shader
- Trace ray through each screen-space fragment coordinate into –z direction



# (Hardware) GPU Ray Tracing




https://developer.nvidia.com/blog/introduction-nvidia-rtx-directx-ray-tracing/