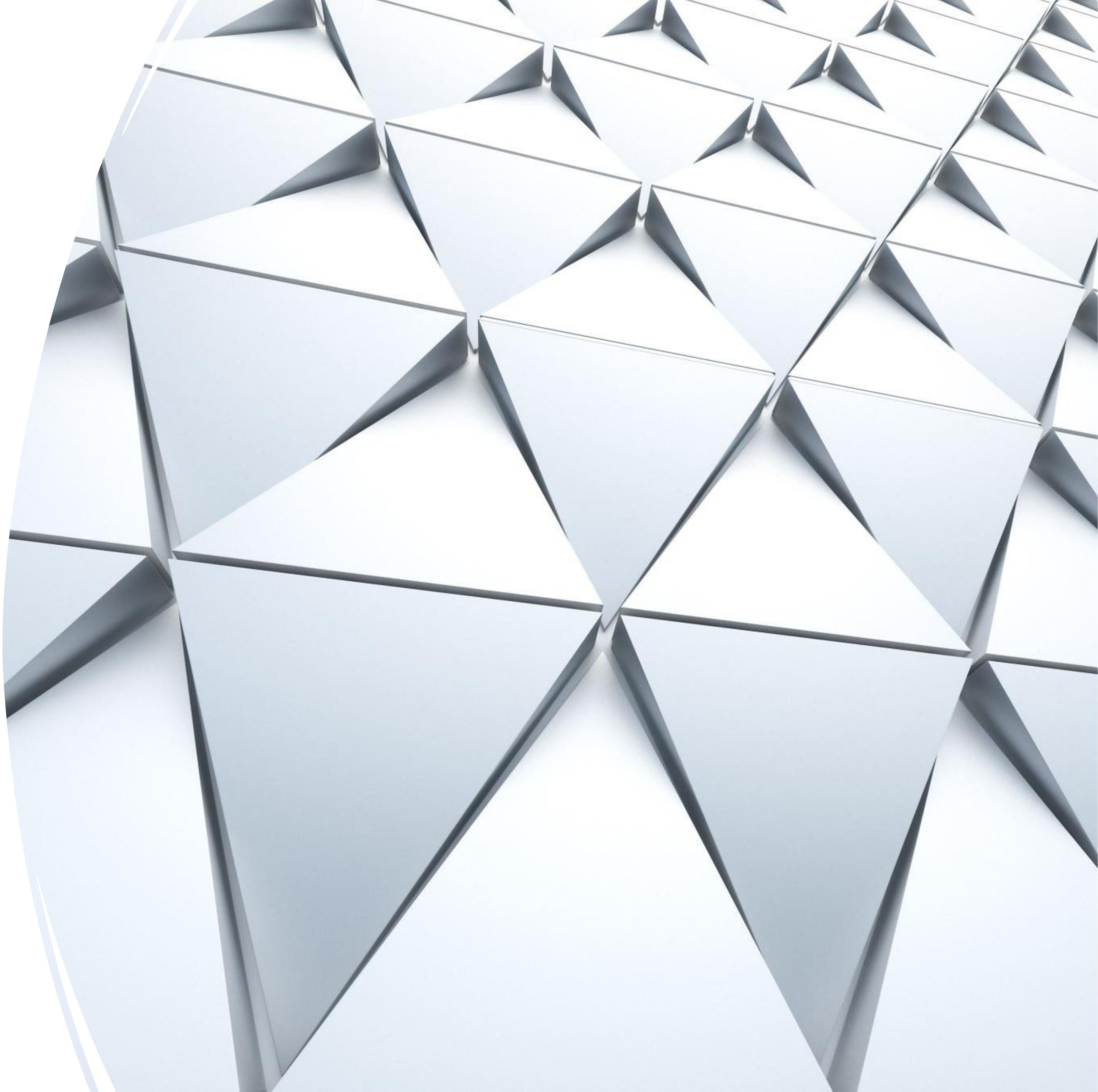
Ray Tracing in Entertainment Industry

Tanaboon Tongbuasirilai
Dept. Computer Science
Kasetsart University

Week 4
Basic shapes and intersections

Basic shapes and intersections

- Sphere
- Quad
- Triangle (not available)



More ray tracing components (classes)

- Scene class
  - Handle scene objects and events.
- Integrator class
  - Solve the rendering equation.
- Material class
  - Fancy color ! - make objects colorful.
- Object class
  - Implement shapes and their ray-intersection method.

Scene class

- Hold objects (sphere, quad, triangle, etc.) to setup a scene
- Attributes
  - A list of objects
  - A list of hit objects
- Methods
  - Add an object
  - Find an intersection by given a ray shot in the scene
  - Get a list of hit objects

Method : find_intersection()

Input

- A generated ray from the camera.
- An interval of parameter 't'.

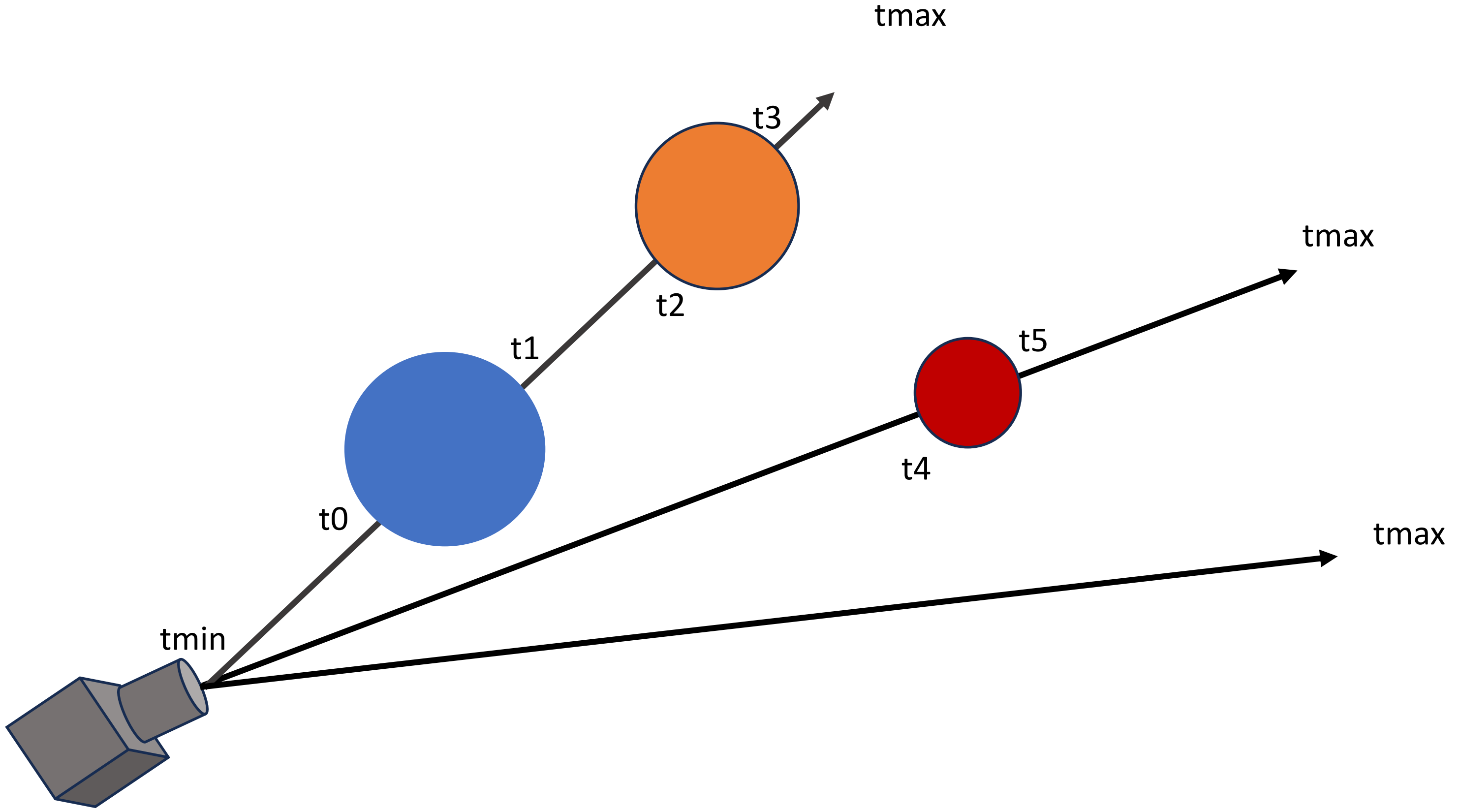
Pseudocode

- Initialize the closet maximum of t.
- For each object in the given scene.
  - Get the hit info from the intersection between an object and the given ray.
  - If the object is hit by the given ray.
    - Update the closet maximum of t.
    - Update the hit list.
  - Return if found any hit or not.

Illustration of the method

Pseudocode

- Initialize the closet maximum of t.
- For each object in the given scene.
    - Get the hit info from the intersection between an object and the given ray.
    - If the object is hit by the given ray.
        - Update the closet maximum of t.
        - Update the hit list.
- Return if found any hit or not.



Integrator class

- Compute the radiance information of a generated ray given the scene.
- Attributes
  - None
- Methods
  - Compute scattering

Method : compute_scattering()

- Input
  - A generated ray
  - The scene

- Pseudocode
  - If the generated ray hits an object.
    - Get the hit info.
    - Get the material of the object.
    - Return the color.

Material class

- Appearance information (color and how the ray interacts with the object.)
- It is a base class.
- Attributes
    - None
- Methods
    - Virtual method : scattering()

A simple material

- Lambertian class
- It is derived from the 'material' class.
- It returns the color of the object.
- Attributes
  - Object's color --> called albedo
- Methods
  - Scattering()

Creating an object

Define an object base class

- Attributes
  - None – each children has its own parameters.
- Methods
  - Virtual method : intersect() - for handling ray-object intersection

The object class is then derived to child classes.

- Sphere
- Quad
- Triangle
- etc.

Object class

An interface class to be derived.

It contains the following virtual methods.

- intersect()

Sphere class

- Attributes
  - Center
  - Radius
  - Material
- Methods
  - Add material
  - intersect()

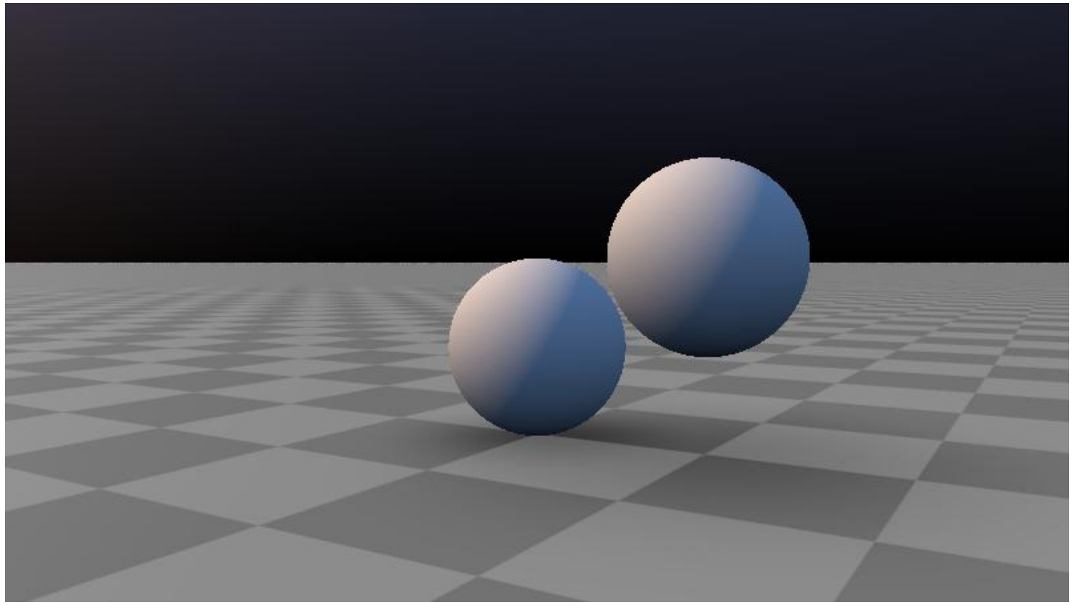


Image from www.shadertoy.com/view/ltBSzK

How a ray intersect a sphere ?

$$(x - C_x)^2 + (y - C_y)^2 + (z - C_z)^2 = r^2$$

Redefining the implicit form of sphere.

$$(\mathbf{P} - \mathbf{C}) \cdot (\mathbf{P} - \mathbf{C}) = r^2$$

A point P on sphere with Center C

$$((\mathbf{A} + t\mathbf{b}) - \mathbf{C}) \cdot ((\mathbf{A} + t\mathbf{b}) - \mathbf{C}) = r^2$$

$$t^2\mathbf{b} \cdot \mathbf{b} + 2t\mathbf{b} \cdot (\mathbf{A} - \mathbf{C}) + (\mathbf{A} - \mathbf{C}) \cdot (\mathbf{A} - \mathbf{C}) - r^2 = 0$$

Look familiar ?

Quadratic solution : $\frac{-b \pm \sqrt{b^2 - 4ac}}{2a}$

$$a = \mathbf{b} \cdot \mathbf{b}$$

$$b = 2\mathbf{b} \cdot (\mathbf{A} - \mathbf{C})$$

$$c = (\mathbf{A} - \mathbf{C}) \cdot (\mathbf{A} - \mathbf{C}) - r^2$$

Equations from raytracing.github.io/books/RayTracingInOneWeekend.html

Roots of the ray-sphere intersection

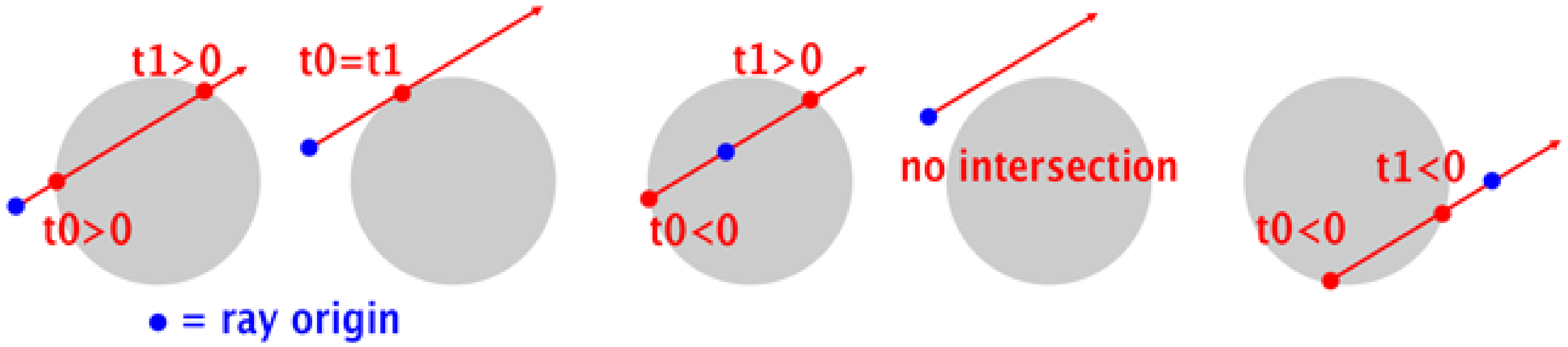


Image from www.scratchapixel.com/lessons/3d-basic-rendering/minimal-ray-tracer-rendering-simple-shapes/ray-sphere-intersection.html

Quadrilateral

- Definition
  - A quadrilateral, sometimes also known as a tetragon or quadrangle is a **four-sided polygon**. If not explicitly stated, all four polygon vertices are generally taken to lie in a plane. (If the points do not lie in a plane, the quadrilateral is called a skew quadrilateral.)
- Implicit form of a plane

$$Ax + By + Cz = D$$

Equation from mathworld.wolfram.com/Quadrilateral.html

Quadrilateral (Plane) representations

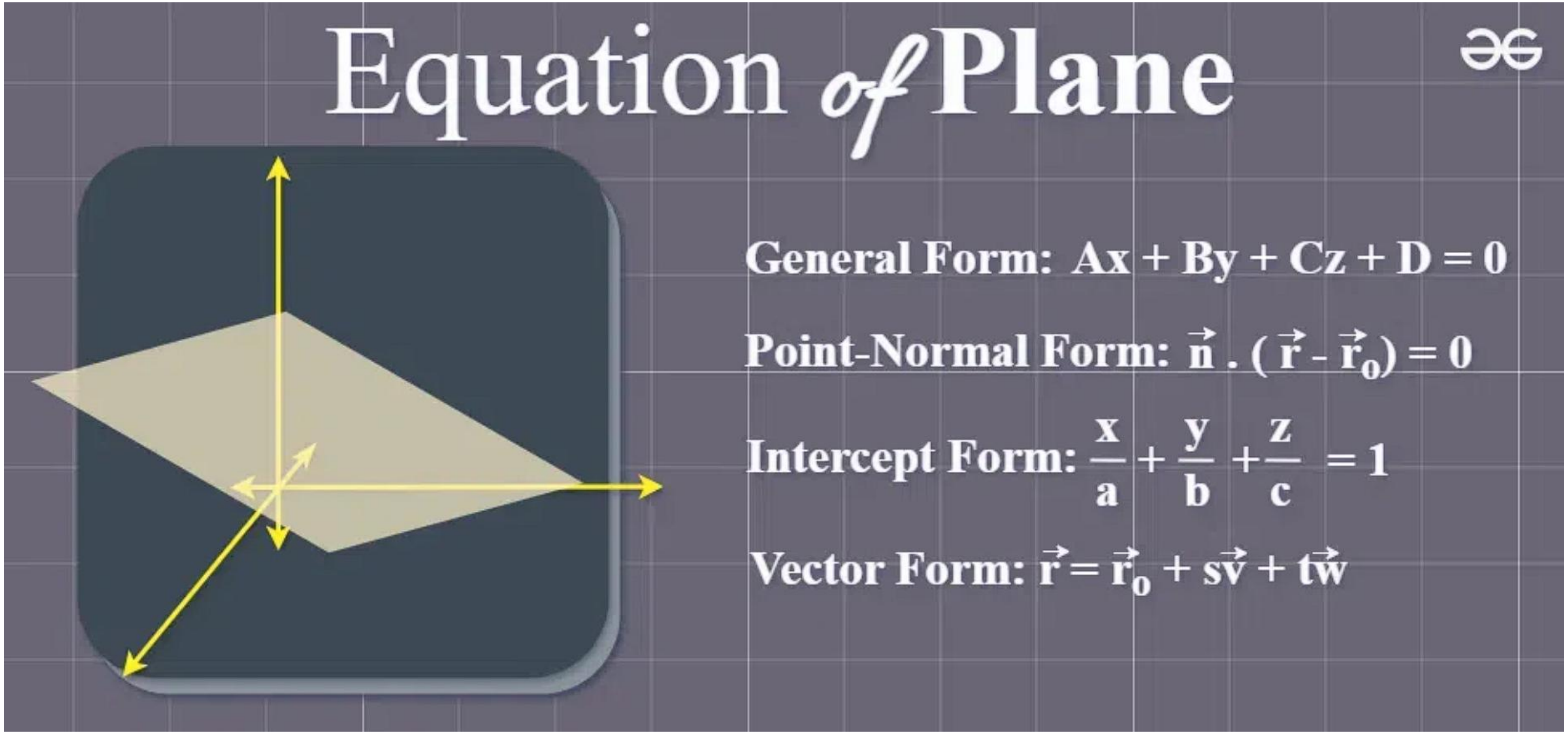


Image from https://www.geeksforgeeks.org/maths/equation-of-plane/

Ray-plane intersection

$$Ax + By + Cz = D$$

Given **n** is the normal perpendicular to the plane.

And **v** is any point lying on the plane.

$$\mathbf{n} \cdot \mathbf{v} = D$$

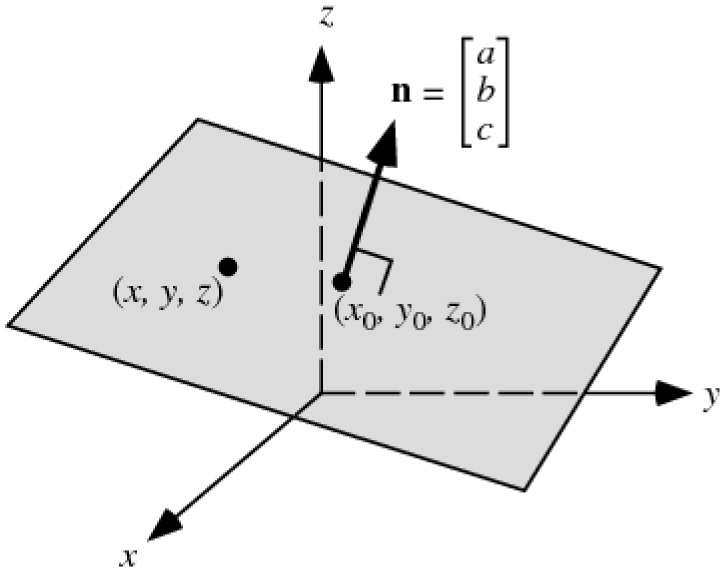
$$\mathbf{n} \cdot (\mathbf{P} + t\mathbf{d}) = D$$

Substitute ray equation

$$\mathbf{n} \cdot \mathbf{P} + \mathbf{n} \cdot t\mathbf{d} = D$$

$$\mathbf{n} \cdot \mathbf{P} + t(\mathbf{n} \cdot \mathbf{d}) = D$$

$$t = \frac{D - \mathbf{n} \cdot \mathbf{P}}{\mathbf{n} \cdot \mathbf{d}}$$

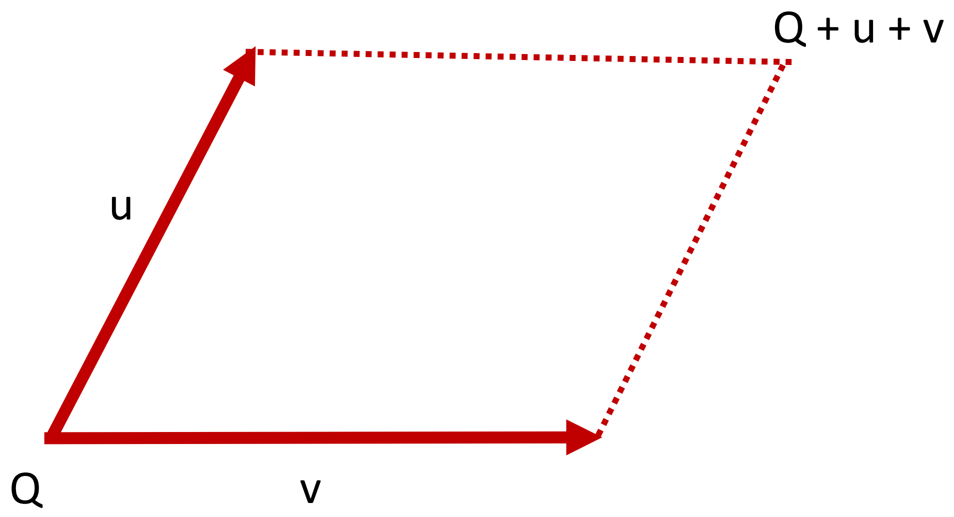


Solution where any parametric ray intersects the plane.

Image from http://www.gtmath.com/2015/04/the-plane-in-r3.html.

Defining a quad

- Q = a corner of the quad
- u, v are vectors forming two sides of the quad



Substituting values

$$\mathbf{n} \cdot \mathbf{P} + \mathbf{n} \cdot t\mathbf{d} = D$$

$$\mathbf{n} \cdot \mathbf{P} + t(\mathbf{n} \cdot \mathbf{d}) = D$$

$$t = \frac{D - \mathbf{n} \cdot \mathbf{P}}{\mathbf{n} \cdot \mathbf{d}}$$

What is n (normal) ?

$$\mathbf{n} = \text{unit\_vector}(\mathbf{u} \times \mathbf{v})$$

What is D ?

$$D = n_x Q_x + n_y Q_y + n_z Q_z$$
$$= \mathbf{n} \cdot \mathbf{Q}$$

What if the denominator is zero ?

Division by zero is undefined. That means the ray is parallel to the plane.

Quad class

Attributes

- A point at a corner
- Vectors representing two sides from the defined corner
- Material
- Normal of the quad

Methods

- Add material
- intersect()

What is the ray-quad intersection solution ?

- Adding a constraint to the solution.
- A quad can be defined by adding boundaries of the plane.
- One simple constraint is as follows.

$$\mathbf{P} = \mathbf{Q} + \alpha\mathbf{u} + \beta\mathbf{v}$$

How do alpha and beta parameters have impacts to the quad constraint ?

Illustration of the constraint

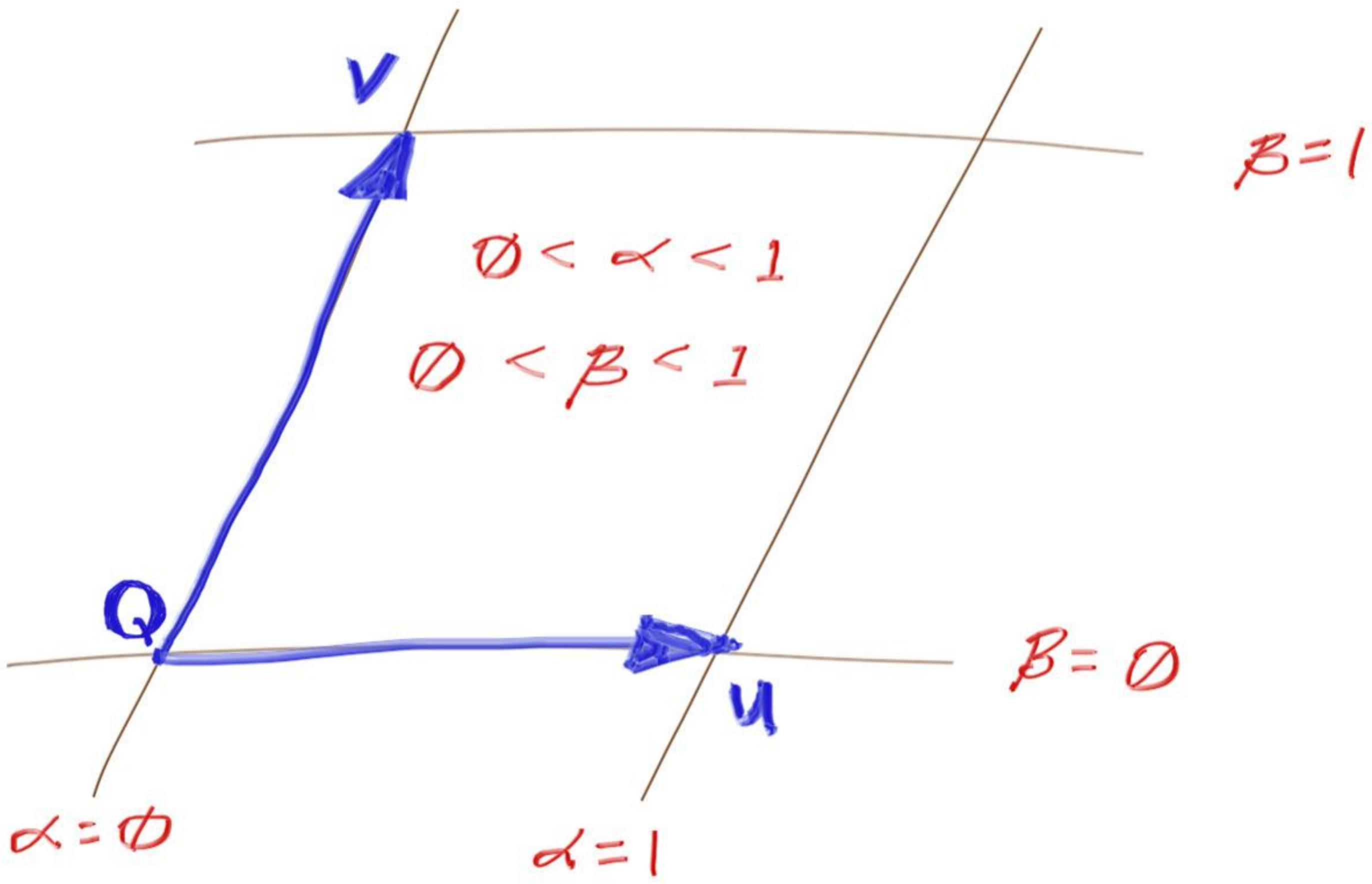


Image from raytracing.github.io/books/RayTracingTheNextWeek.html

Calculating alpha and beta parameters

These normal are UxV.

$$\mathbf{w} = \frac{\mathbf{n}}{\mathbf{n} \cdot (\mathbf{u} \times \mathbf{v})} = \frac{\mathbf{n}}{\mathbf{n} \cdot \mathbf{n}}$$

$$\alpha = \mathbf{w} \cdot (\mathbf{p} \times \mathbf{v})$$

$$\beta = \mathbf{w} \cdot (\mathbf{u} \times \mathbf{p})$$

Codes and class assignment !

- Github : RT-python-week04
  - https://github.com/KUGA-01418283-Raytracing/RT-python-week04