**1 - Why we need Homogeneous Coordinates over Cartesian Coordinates?**

Cartesian Coordinates (x, y) allows simple transformations like rotation, translation ,scaling e.t.c with ease but when it comes to perspective transformations like perspective projection or zooming in camera, it fails badly.  
Homogenous Coordinates introduces a dimension w so that each 2D point is represented as 3D space (x,y,1) , which allows:

1 - Perspective Transformations (Zooming e.t.c)

2 – Handling of vanishing /infinity point.

3 – Handle complex mathematical operations with matrix multiplications.  
For example, a car moving away from you shrinks in size cannot be explained with cartesian coordinates but that new dimension (w) can help to explain it.

O A point in homogeneous coordinates is represented as 𝒑= (𝒙, 𝒚, 𝒘) 𝑻

O A line in homogeneous coordinates is represented as **𝑙** **= (𝑎, 𝑏, 𝑐) 𝑇**

O The equation of a line in homogeneous coordinates is given by: **a 𝒙+ b 𝒚+ c 𝒘= 0**

**2 – Projective Space?**

The set of homogeneous equivalence classes

(HEC) of vectors in ℝ 3- (0, 0, 0) T is called the projective space ℙ𝟐 . It is actually **the set of all unique lines in 2D**

* **Homogeneous Coordinates:** In projective space, a point **p** is represented as (x, y, w) instead of (x, y), introducing an extra dimension w.
* **Points at Infinity:** Parallel lines, which never meet in **Euclidean geometry**, **intersect at infinity** in projective space, solving perspective issues.
* **Simplified Transformations:** Transformations like **rotation, translation, and scaling** become **easier** using matrix multiplication.
* **Multiple Representations:** Scaling a point by any **nonzero scalar** does not change its nature, meaning each point has **infinite equivalent representations**

**3- Cartesian Coordinates and Projective Geometry in Action:**

(3D Graphics, Robotics, CV,AR)

* Helps in 3D scene rendering in 3D Graphics.
* Help in depth detection and object tracking in computer Vision and Robotics.
* Helps in Augmented Reality in placing Virtual objects in real word environment

**4 - If you multiply (x, y, 1) by a scalar k, does it still represent the same point?**

Yes! In homogeneous coordinates, points are defined **up to scale**, meaning multiplying **(x, y,1)** by any **nonzero scalar k** results in **(kx, ky, k)**. This does **not change the actual point**, only its representation. When converted back to Cartesian coordinates, we still get **(x, y)**.

**Key Takeaway:** The point remains the same, but its **homogeneous representation changes.**

**5 – How to check any point p lies on the line?**

* Points are represented as (x, y, w)
* Lines are represented as (a, b, c)
* A point (x, y, 1) lies on a line (a, b, c) if:

ax + by + c = 0.

This is just the dot product of (a, b, c) and (x, y, 1) being zero.

* This process is called **normalization**, as we divide the first two components by the third component to return to **(x, y) in Euclidean space**.
* If a point (x, y, w) lies on a line, then **any scaled version** of the point (kx, ky, kw) for k≠0 will also lie on the same line.

**6 -** **What Are Points at Infinity?**

In projective geometry, not all points correspond to real-world locations in Cartesian space. Some points exist at infinity, meaning they represent directions rather than fixed positions.

Understanding Points at Infinity

🔹 Every 2D point (x, y) in Cartesian space is written in homogeneous coordinates as (x, y, 1).

🔹 Points at infinity occur when w = 0, making them impossible to represent in normal Cartesian space. 🔹 The origin (0,0,0) is excluded because it does not define a meaningful direction—it represents nothingness rather than a location.

**7 -** **Finding the Intersection of Two Lines Using Homogeneous Coordinates**

In homogeneous space, the intersection of two lines is found using the cross product.  
Given two lines in homogeneous form:

* **2x + 3y − 5 →** **l (2, 3, −5) l**
* **4x − y + 2 →** **l (4, −1, 2) l**
* ✔ Compute the cross product p= L1×L2 p ​ to get the intersection point.  
  ✔ Convert the result from homogeneous to Cartesian by dividing x and y by w

**8 – Why origin not included in the projective space?**

* The origin (0, 0, 0) is excluded because it has no direction or unique representation as Scaling doesn’t change the origin.
* Every point in projective space represents a line, and the origin is already part of every line thus it cannot be assigned to a single unique point.
* In cameras, the origin is the pinhole where everything collapses, making it useless for projective geometry. If we included the origin, everything in the scene would collapse into a **single point**, losing all depth and structure

9 - **Duality Between Points and Lines in Projective Geometry**

In **projective geometry**, **points and lines have a special relationship** known as **duality**. This means that the **mathematical representation of a point and a line is similar**, and one can be transformed into the other using basic operations.

If a **point** x = (x1, x2, x3)T lies on a **line** l = (a, b, c), then their **dot product** is **zero**:

xT⋅l = ax1 + bx2+ cx3 = 0

**10 - Why Projective Geometry**?

There are four reasons:

1. Camera is a projective engine

2. Points at infinity are handled

3. Algebra is simpler than usual

4. Is the most general framework to work in Affine or Euclidean upgrades can be made if required.

**11- What is PTZ?**

Pan Tilt Zoom camera is a robotic or surveillance camera which allows remove control on its movement and zooming functionality.

* **PAN** – Horizontal Movement: to cover the wide range view.
* **TILT** – Vertical Movement: to track objects at different heights
* **ZOOM** – Optical and digital: changes focal point to zoom in or zoom out

**Application:**

It is used in: (SIB):

* Security and Surveillance: Its is used for security purposes in road, monitoring traffic or at commercial areas.
* Industry and Search: Used in Automation, robotics and remote monitoring.
* Broadcasting and Live events: It is used for News and sports broadcasting to cover from various angles.