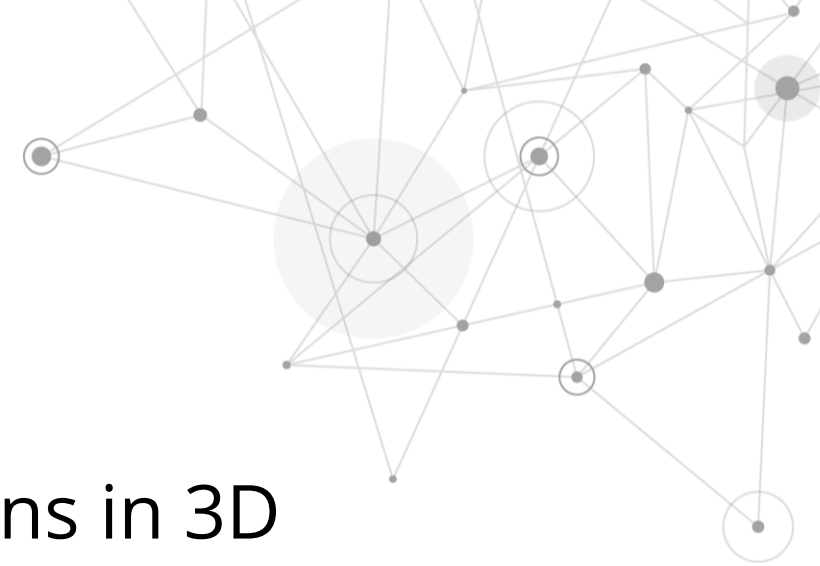
GAME2016
Mathematical Foundation of Game Design and Animation

Lecture 6
Homogeneous Matrices and General Affine Transformations

Dr. Paolo Mengoni
pmengoni@hkbu.edu.hk
Senior Lecturer @HKBU Department of Interactive Media

Agenda



- Homogeneous matrices
  - From vectors to 4×4 matrices
- General Affine Transformations in 3D
- Perspective projection at a glance

4×4 Homogenous Matrices

Why Homogenous Matrices?

- Simplify the management of transformations.
- It will let us handle translation with a matrix transformation.
- Single framework to work with when developing a software system.

Homogenous Coordinates

- Extend 3D into 4D.
- The 4^{th} dimension is not “time”.
- The 4^{th} dimension is really just a trick to help the math work out
- The 4^{th} dimension is called *w*.

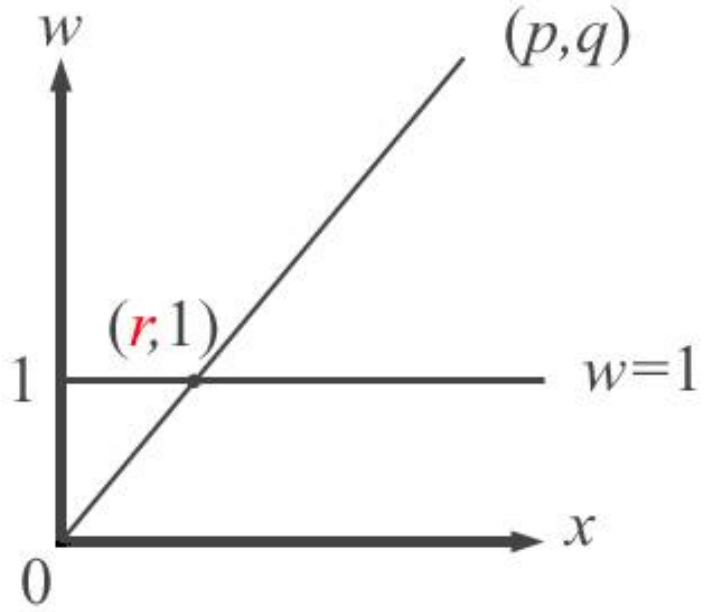
Extending 1D into Homogenous Space

- Start with 1D, its easier to visualize than 3D.
- Homogenous 1D coords are of the form (x, w).
- Imagine the vanilla 1D line lying at $w = 1$.
- So the 1D point x has homogenous coords (x, 1).
- Given a homogenous point (x, w), the corresponding 1D point is its projection onto the line $w = 1$ along a line to the origin, which turns out to be (x/w, 1).

Projecting Onto 1D Space

- Each point *x* in 1D space corresponds to an infinite number of points in homogenous space, those on the line from the origin through the point (*x*, 1).

- The homogenous points on this line project onto its intersection with the line *w* = 1.



Simultaneous Equations

- Equation of line is $w = ax + b$
- (p, q) and $(0, 0)$ are on the line.
- Therefore:

$$q = ap + b$$
$$0 = a0 + b,$$

- That is, $b = 0$ and $a = q/p$.

What is *r*?

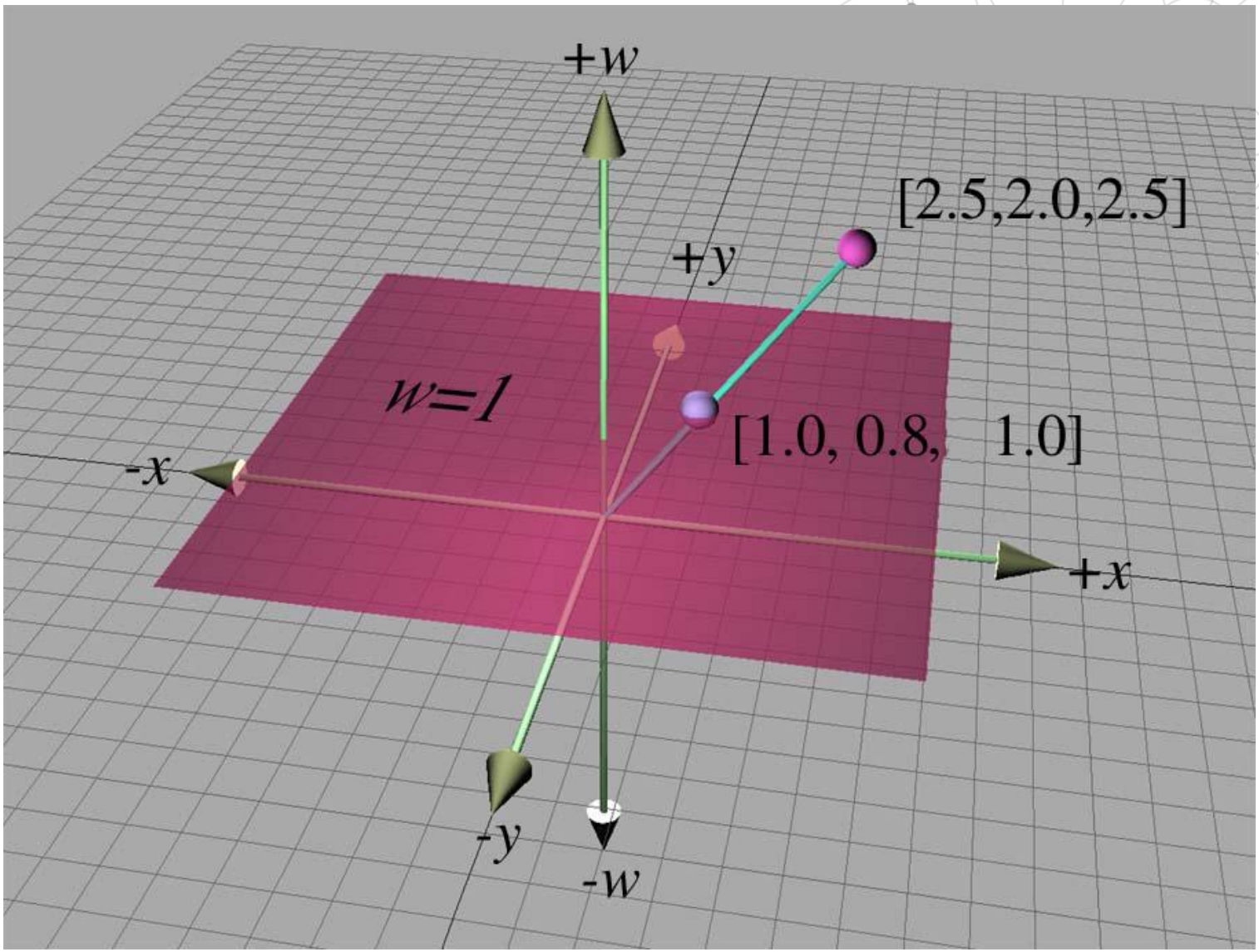
- The line equation is $w = qx/p$.
- Therefore, when $w = 1$, $x = p/q$.
- This means that $r = p/q$.
- The homogenous point (p, q) projects onto the 1D point $(p/q, 1)$.
- That is, the 1D equivalent of the homogenous point (p, q) is p/q.

Extending 2D into Homogenous Space

- 2D next, it's *still* easier to visualize than 3D.
- Homogenous 2D coordinates are of the form (x, y, w).
- Imagine the vanilla 2D plane lying at $w = 1$.
- So the 2D point (x, y) has homogenous coordinates (x, y, 1).

Projecting Onto 2D Space

- Each point (x, y) in 2D space corresponds to an infinite number of points in homogenous space.
- Those on the line from the origin thru (x, y, 1).
- The homogenous points on this line project onto its intersection with the plane $w = 1$.


+w
[2.5,2.0,2.5]
+y
w=1
[1.0, 0.8, 1.0]
-x
+x
-y
-w

2D Homogenous Coordinates

- Just like before (argument omitted), the homogenous point (*x*, *y*, *w*) corresponds to the 2D point (*x/w*, *y/w*, 1).

- That is, the 2D equivalent of the homogenous point (*p*, *q*, *r*) is (*p/r*, *q/r*).

3D Homogenous Coordinates

- This extends to 3D in the obvious way.
- The homogenous point (*x, y, z, w*) corresponds to the 3D point (*x/w, y/w, z/w,* 1).
- That is, the 3D equivalent of the homogenous point (*p, q, r, s*) is (*p/s, q/s, r/s*).

Point at Infinity

- *w* can be any value except 0 (divide by zero error).
- The point (*x*,*y*,*z*,0) can be viewed as a “point at infinity”

Why Use Homogenous Space?

- It will let us handle translation with a matrix transformation.
- Embed 3D space into homogenous space by basically ignoring the *w* component.
- Vector (*x, y, z*) gets replaced by (*x, y, z*, 1).
- Does that “1” at the end sound familiar?

Homogenous Matrices

- Embed 3D transformation matrix into 4D matrix by using the identity in the *w* row and column.

$$\begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix} \Longrightarrow \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 \\ m_{21} & m_{22} & m_{23} & 0 \\ m_{31} & m_{32} & m_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D Matrix Multiplication

$$\begin{bmatrix} x & y & z \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} \\ m_{21} & m_{22} & m_{23} \\ m_{31} & m_{32} & m_{33} \end{bmatrix}$$

$$= \begin{bmatrix} xm_{11} + ym_{21} + zm_{31} \\ xm_{12} + ym_{22} + zm_{32} \\ xm_{13} + ym_{23} + zm_{33} \end{bmatrix}^T$$

4D Matrix Multiplication

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} m_{11} & m_{12} & m_{13} & 0 \\ m_{21} & m_{22} & m_{23} & 0 \\ m_{31} & m_{32} & m_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} xm_{11} + ym_{21} + zm_{31} \\ xm_{12} + ym_{22} + zm_{32} \\ xm_{13} + ym_{23} + zm_{33} \\ 1 \end{bmatrix}^T$$

Translation Matrices

3D translation matrix by shearing 4D space.

$$\begin{bmatrix} x & y & z & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \Delta x & \Delta y & \Delta z & 1 \end{bmatrix}$$

$$= \begin{bmatrix} x + \Delta x & y + \Delta y & z + \Delta z & 1 \end{bmatrix}$$

Translation vs. Orientation

- Just like in 3D, compose 4D operations by multiplying the corresponding matrices.
- The translation and orientation parts of a composite matrix are independent.
- For example, let **R** be a rotation matrix and **T** be a translation matrix.
- What does **M** = **RT** look like?

$$\mathbf{R} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \Delta x & \Delta y & \Delta z & 1 \end{bmatrix}$$

Rotate then Translate

- Then we could rotate and then translate a point **v** to a new point **v′** using **v′** = **vRT**.
- We are rotating first and then translating.
- The order of transformations is important, and since we use row vectors, the order of transformations coincides with the order that the matrices are multiplied, from left to right.

$\mathbf{M} = \mathbf{RT}$

- Just as with 3 x 3 matrices, we can concatenate the two matrices into a single transformation matrix, which we'll call **M**.

Let $\mathbf{M} = \mathbf{RT}$, so

$$\mathbf{v}' = \mathbf{vRT} = \mathbf{v}(\mathbf{RT}) = \mathbf{vM}$$

$$\mathbf{M} = \mathbf{RT}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ \Delta x & \Delta y & \Delta z & 1 \end{bmatrix}$$

$$= \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ \Delta x & \Delta y & \Delta z & 1 \end{bmatrix}$$

In Reverse

- Applying this information in reverse, we can take a 4 x 4 matrix **M** and separate it into a linear transformation portion, and a translation portion.

- We can express this succinctly by letting the translation vector $\mathbf{t} = [\Delta x, \Delta y, \Delta z]$.

$$\mathbf{M} = \begin{bmatrix} \mathbf{R} & \mathbf{0} \\ \mathbf{t} & 1 \end{bmatrix}$$

Points at Infinity Again

- Points at infinity are actually useful.
- They orient just like points with $w \neq 0$: multiply by the orientation matrix.
- But they don't translate: translation matrices have no effect on them.

Matrix Without Translation

$$\begin{bmatrix} x & y & z & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} xr_{11} + yr_{21} + zr_{31} \\ xr_{12} + yr_{22} + zr_{32} \\ xr_{13} + yr_{23} + zr_{33} \\ 0 \end{bmatrix}^T$$

Matrix With Translation

$$\begin{bmatrix} x & y & z & 0 \end{bmatrix} \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ \Delta x & \Delta y & \Delta z & 1 \end{bmatrix}$$

$$= \begin{bmatrix} xr_{11} + yr_{21} + zr_{31} \\ xr_{12} + yr_{22} + zr_{32} \\ xr_{13} + yr_{23} + zr_{33} \\ 0 \end{bmatrix}^T$$

Same

So What?

- The translation part of 4D homogenous transformation matrices has no effect on points at infinity.

- Use points at infinity for things that don't need translating (eg. Surface normals).

- Use regular points (with $w = 1$) for things that *do* need translating (eg. Points that make up game objects).

General Affine Transformations

General Affine Transformations

- Armed with 4 x 4 transform matrices, we can now create more general affine transformations that contain translation.

- For example:
  - Rotation about an axis that does not pass through the origin
  - Scale about a plane that does not pass through the origin
  - Reflection about a plane that does not pass through the origin
  - Orthographic projection onto a plane that does not pass through the origin

General Affine Transformations

- Basic idea:
  - Translate the center of the transformation to the origin
  - Perform the linear transformation using the techniques developed in previous lectures
  - Transform the center back to its original location

- We will start with a translation matrix **T** that translates the point **p** to the origin, and a linear transform matrix **R** that performs the linear transformation.

- The final transformation matrix **A** will be the equal to the matrix product $\mathbf{TRT}^{-1}$.
  - $\mathbf{T}^{-1}$ is of course the translation matrix with the opposite translation amount as **T**, not really an inverse matrix

$$\mathbf{T} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ -p_x & -p_y & -p_z & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{p} & 1 \end{bmatrix}$$

$$\mathbf{R}_{4\times4} = \begin{bmatrix} r_{11} & r_{12} & r_{13} & 0 \\ r_{21} & r_{22} & r_{23} & 0 \\ r_{31} & r_{32} & r_{33} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix}$$

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ p_x & p_y & p_z & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{I} & 0 \\ \mathbf{p} & 1 \end{bmatrix}$$

$$\mathbf{T}\mathbf{R}_{4\times4}\mathbf{T}^{-1} = \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ -\mathbf{p} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{0} \\ \mathbf{0} & 1 \end{bmatrix} \begin{bmatrix} \mathbf{I} & \mathbf{0} \\ \mathbf{p} & 1 \end{bmatrix} = \begin{bmatrix} \mathbf{R}_{3\times3} & \mathbf{0} \\ -\mathbf{p}\left(\mathbf{R}_{3\times3}\right) + \mathbf{p} & 1 \end{bmatrix}$$

Observation

- The extra translation in an affine transformation only changes the last row of the 4 x 4 matrix.
- The upper 3 x 3 portion, which contains the linear transformation, is not affected.
- The use of homogenous coordinates so far has really been nothing more than a mathematical trick to allow us to include translation in our transformations.

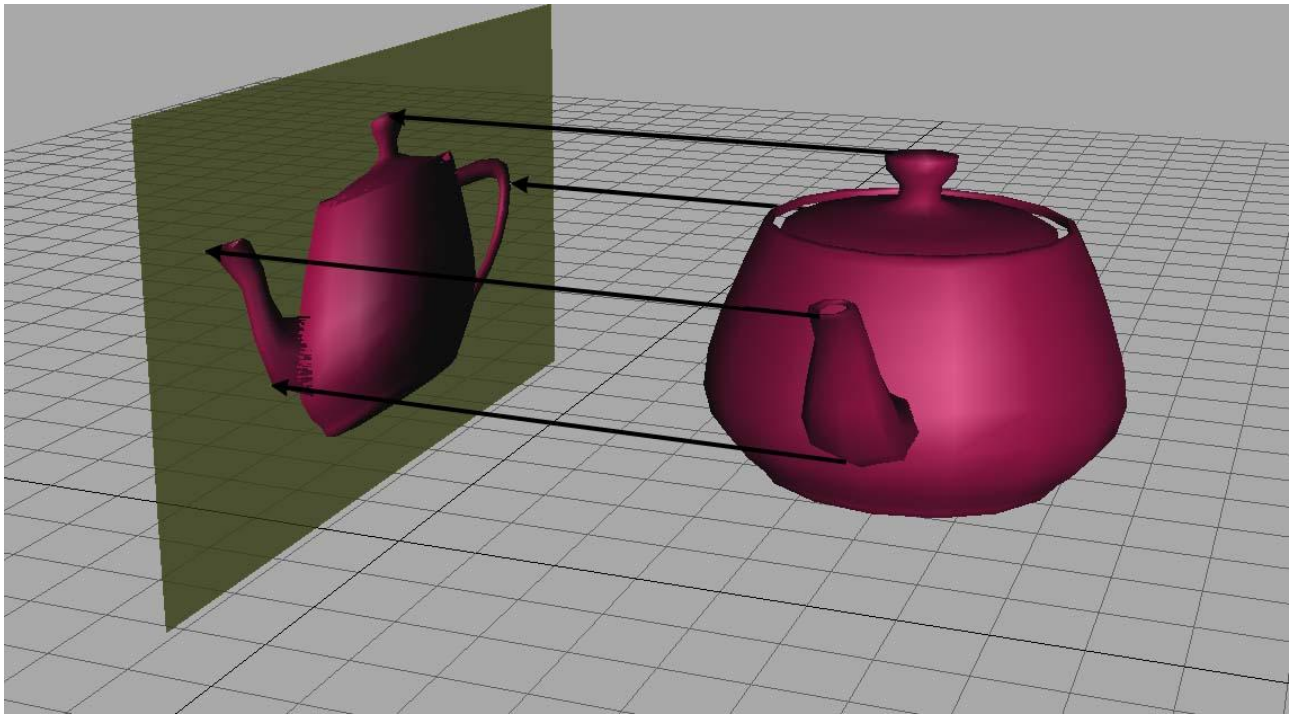
Perspective Projection

Projections

- We've only used $w = 1$ and $w = 0$ so far.
- There's a use for the other values of w too.
- We've seen how to do *orthographic projection* before.
- Now we'll see how to do *perspective projection* too.

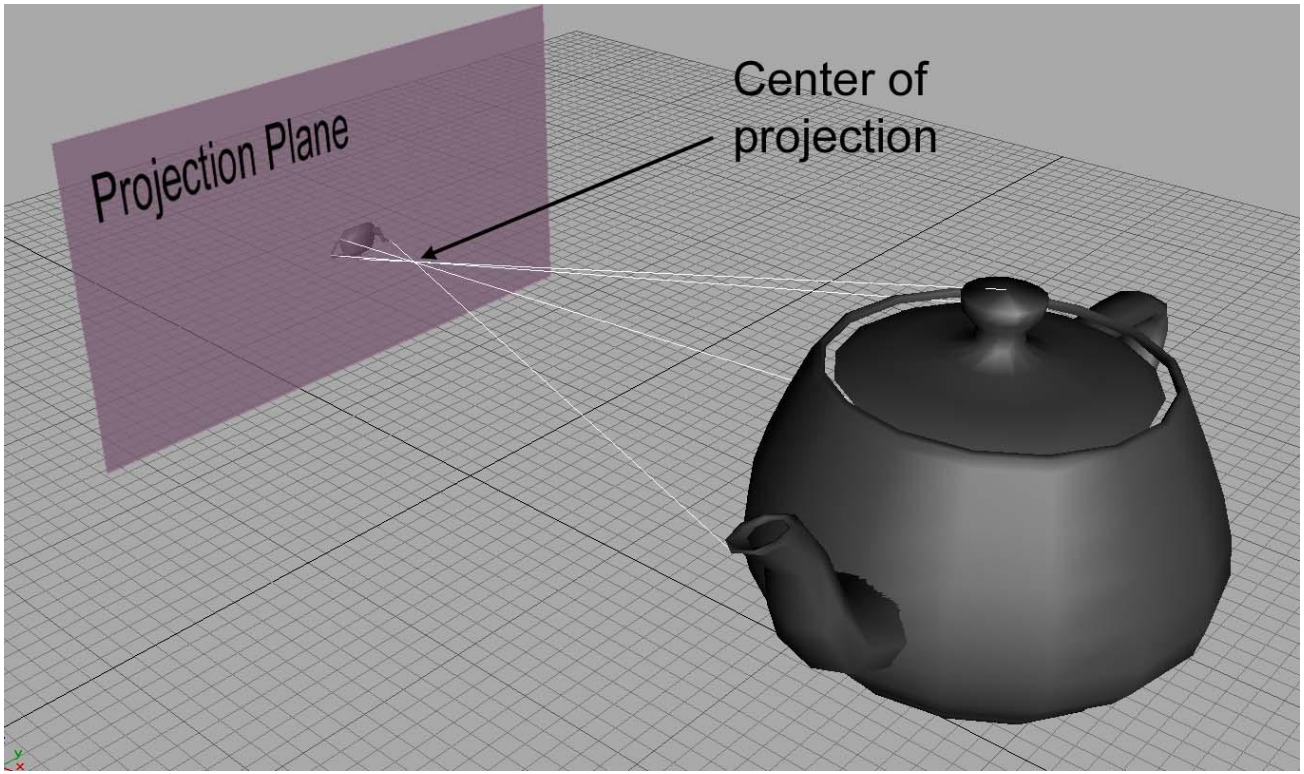
Orthographic Projection

- Orthographic projection has *parallel projectors*.
- The projected image is the same size no matter how far the object is from the projection plane.



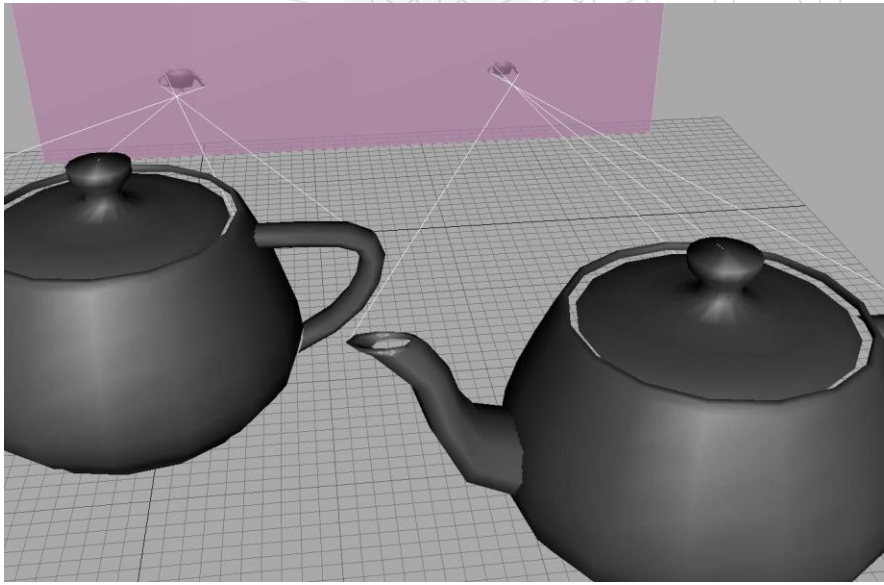
Perspective Projection

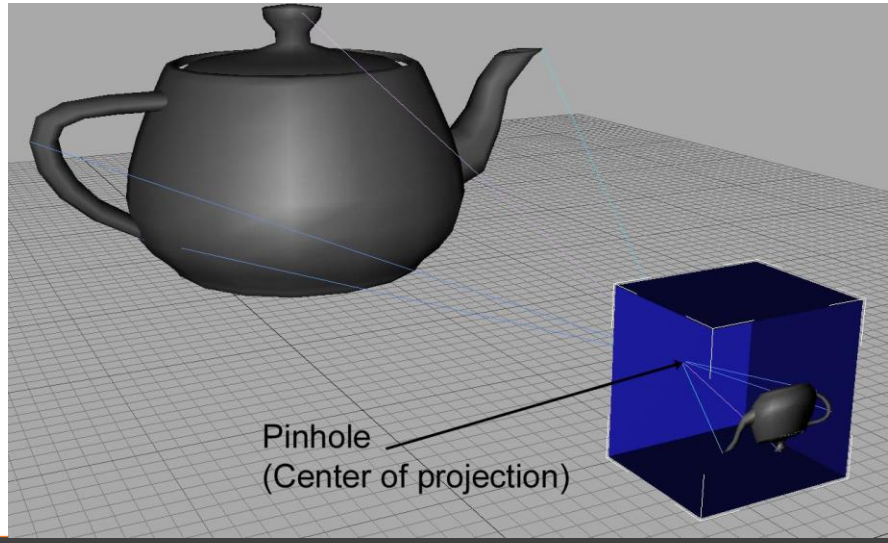
- We want objects to get smaller with distance.



Perspective Foreshortening

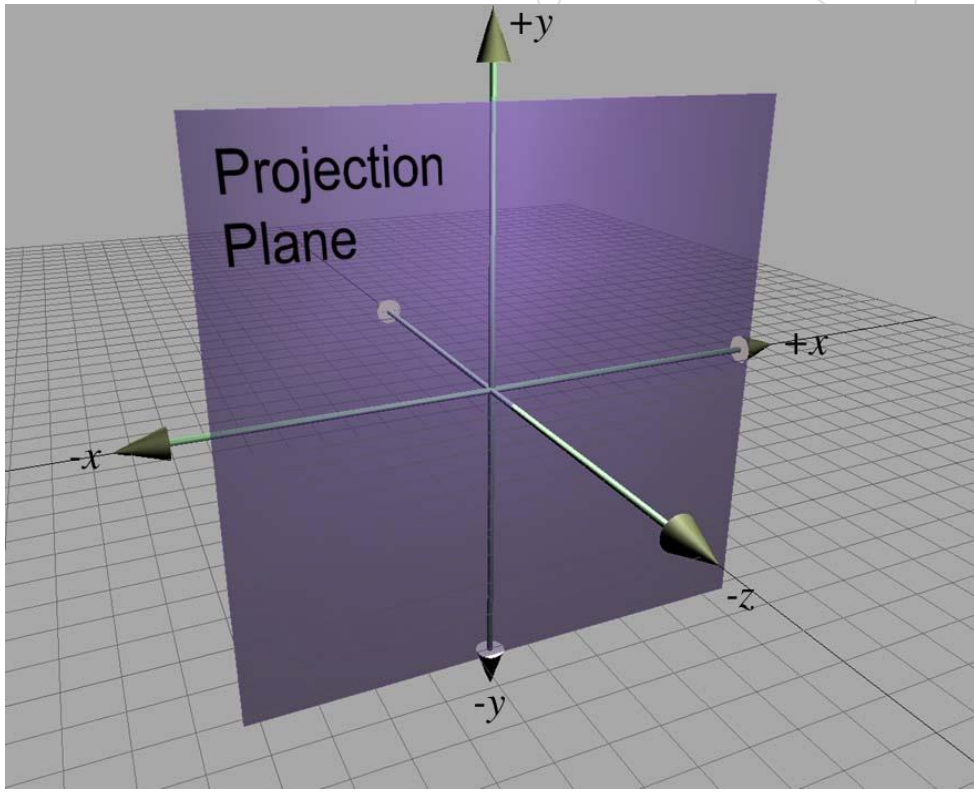
- This is known as *perspective foreshortening*.
- The math is based on a *pinhole camera*.
  - Take a closed box that's very dark inside.
  - Make a pinhole.
- If you point the pinhole at something bright, an image of the object will be projected onto the back of the box.
  - That's kind of how the human eye works too.





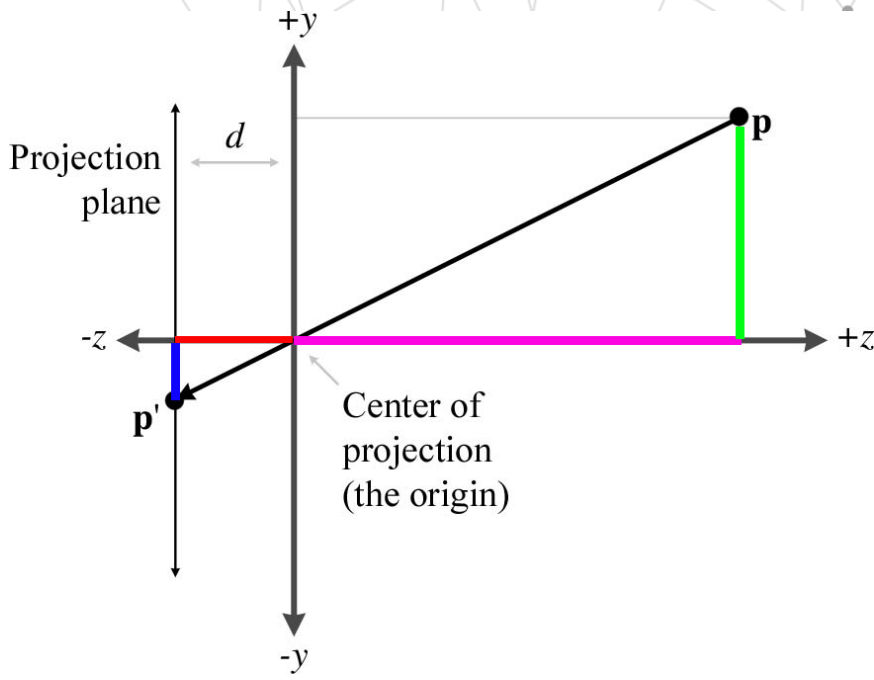
Projection Geometry

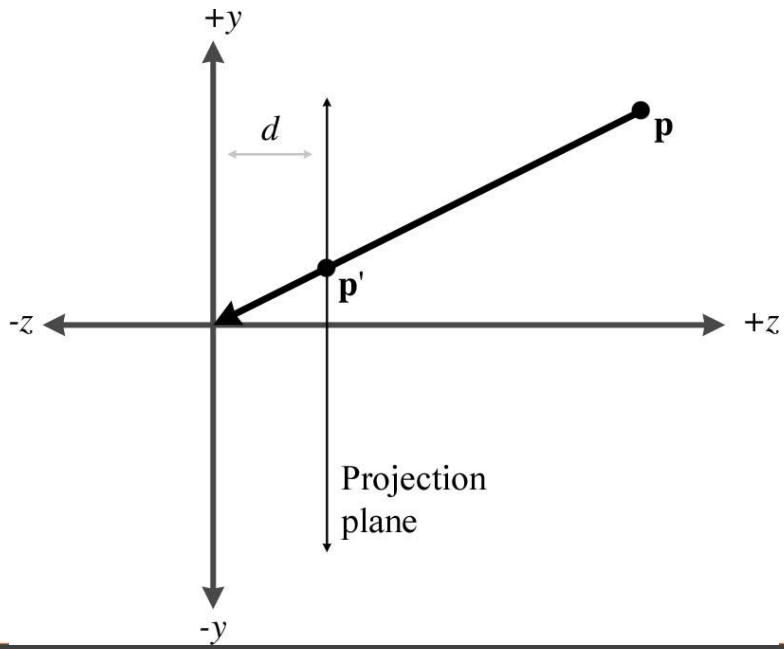
- Let's project on a plane parallel to the *x-y* plane.
- Choose a distance *d* from the pinhole to the projection plane, called the *focal distance.*
- The pinhole is called the *focal point*.
- Put the focal point at the origin and the projection plane at $z = -d$.
  - Remember the concept of *camera space*?



Do the Math

- View it from the side.
- The math can be simplified by shifting the projection plane in front of the focal point.
- Use a 4D homogenous matrix to perform the projection





Why use homogeneous space?

- Homogenous space is interesting
  - 4 x 4 matrices provide a way to express projection as a transformation that can be concatenated with other transformations.
  - Projection onto non-axially aligned planes is possible.

- We may not need homogenous coordinates

BUT

- 4 x 4 matrices provide a compact way to represent and manipulate projection transformations.

Real Projection Matrices

- The projection matrix in a real graphics geometry pipeline is also known as the *clip matrix*

- It has some differences with the method we presented:
  1. It will apply a normalizing scale factor such that $w = 1$ at the far clip plane. Values used for depth buffering will be distributed appropriately and maximize the precision.
  2. The projection matrix in most graphics systems scales all the coordinates' values according to the field of view of the camera.