Topics

1. Introduction: What is Computer Graphics?
2. Raster Images (image input/output devices and representation)
3. Scan conversion (pixels, lines, triangles)
4. Ray Casting (camera, visibility, normals, lighting, Phong illumination)
5. Ray Tracing (shadows, supersampling, global illumination)
6. Spatial Data Structures (AABB trees, OBB, bounding spheres, octree)
7. Meshes (connectivity, smooth interpolation, uv-textures, subdivision, Laplacian smoothing)
8. 2D/3D Transformations (Translate, Rotate, Scale, Affine, Homography, Homogeneous coordinates)
9. Viewing and Projection (matrix composition, perspective, Z-buffer)
10. Shader Pipeline (Graphics Processing Unit)
11. Animation (kinematics, keyframing, Catmull-Romm interpolation, physical simulation)
12. 3D curves and objects (Hermite, Bezier, cubic curves, curve continuity, extrusion/revolve surfaces)
13. Advanced topics overview

Topic 8.

2D/3D Transformations

Transformations

Transformation/Deformation in Graphics:

A function f, mapping points to points.
simple transformations are usually invertible.

$$[x\ y]^T \underset{f^{-1}}{\overset{f}{\rightleftarrows}} [x'\ y']^T$$

Applications:

- Placing objects in a scene.
- Composing an object from parts.
- Animating objects.

Processing Tree Demo!
https://processing.org/examples/tree.html

Lets start out simple...

Translate a point $[x\ y]^T$ by $[t_x\ t_y]^T$:

$$x' = x + t_x$$
$$y' = y + t_y$$

Rotate a point $[x\ y]^T$ by an angle t :

$$x' = x \cos t - y \sin t$$
$$y' = x \sin t + y \cos t$$

Scale a point $[x\ y]^T$ by a factor $[s_x\ s_y]^T$

$$x' = x\, s_x$$
$$y' = y\, s_y$$

Representing 2D transforms as a 2x2 matrix

Rotate a point $[x\ y]^T$ by an angle t :

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} \cos t & -\sin t \\ \sin t & \cos t \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Scale a point $[x\ y]^T$ by a factor $[s_x\ s_y]^T$

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

Translate ?

Cartesian ⇔ Homogeneous 2D Points

Cartesian $[x\ y]^T$ => Homogeneous $[x\ y\ 1]^T$

Homogeneous $[x\ y\ w]^T$ => Cartesian $[x/w \quad y/w \quad 1]^T$

Homogeneous points are equal if they represent the same Cartesian point. For eg. $[4\ {-6}\ 2]^T = [-6\ 9\ {-3}]^T$.

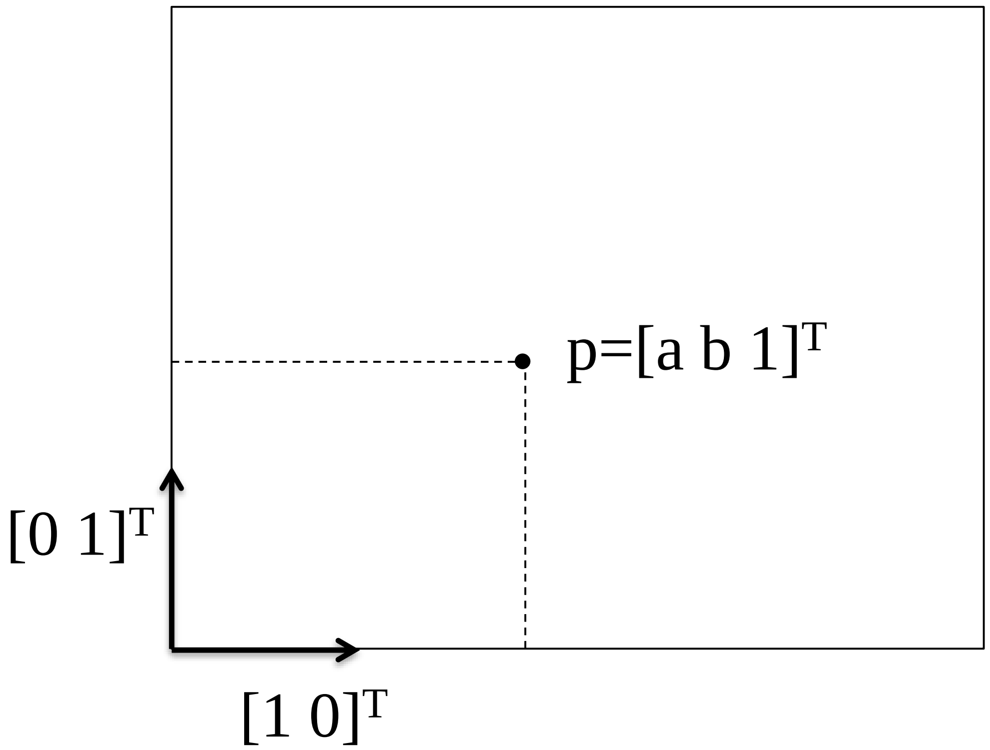
What about w=0?

Points at ∞ in Homogeneous Coordinates

$[x\ y\ w]^T$ with $w=0$ represent points at infinity, though with direction $[x\ y]^T$ and thus provide a natural representation for **vectors**, distinct from **points** in Homogeneous coordinates.

w=0: vector
w!=0: points

Points as Homogeneous 2D Point Coords



$$p = a*[1\ 0\ 0]^T + b*[0\ 1\ 0]^T + [0\ 0\ 1]^T$$

basis vectors

origin

Representing 2D transforms as a 3x3 matrix

Translate a point $[x\ y]^T$ by $[t_x\ t_y]^T$:

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Rotate a point $[x\ y]^T$ by an angle t :

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos t & -\sin t & 0 \\ \sin t & \cos t & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Scale a point $[x\ y]^T$ by a factor $[s_x\ s_y]^T$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Properties of 2D transforms

…these 3x3 transforms have a variety of properties. most generally they map **lines** to **lines**. Such invertible transforms are also called **Homographies**.

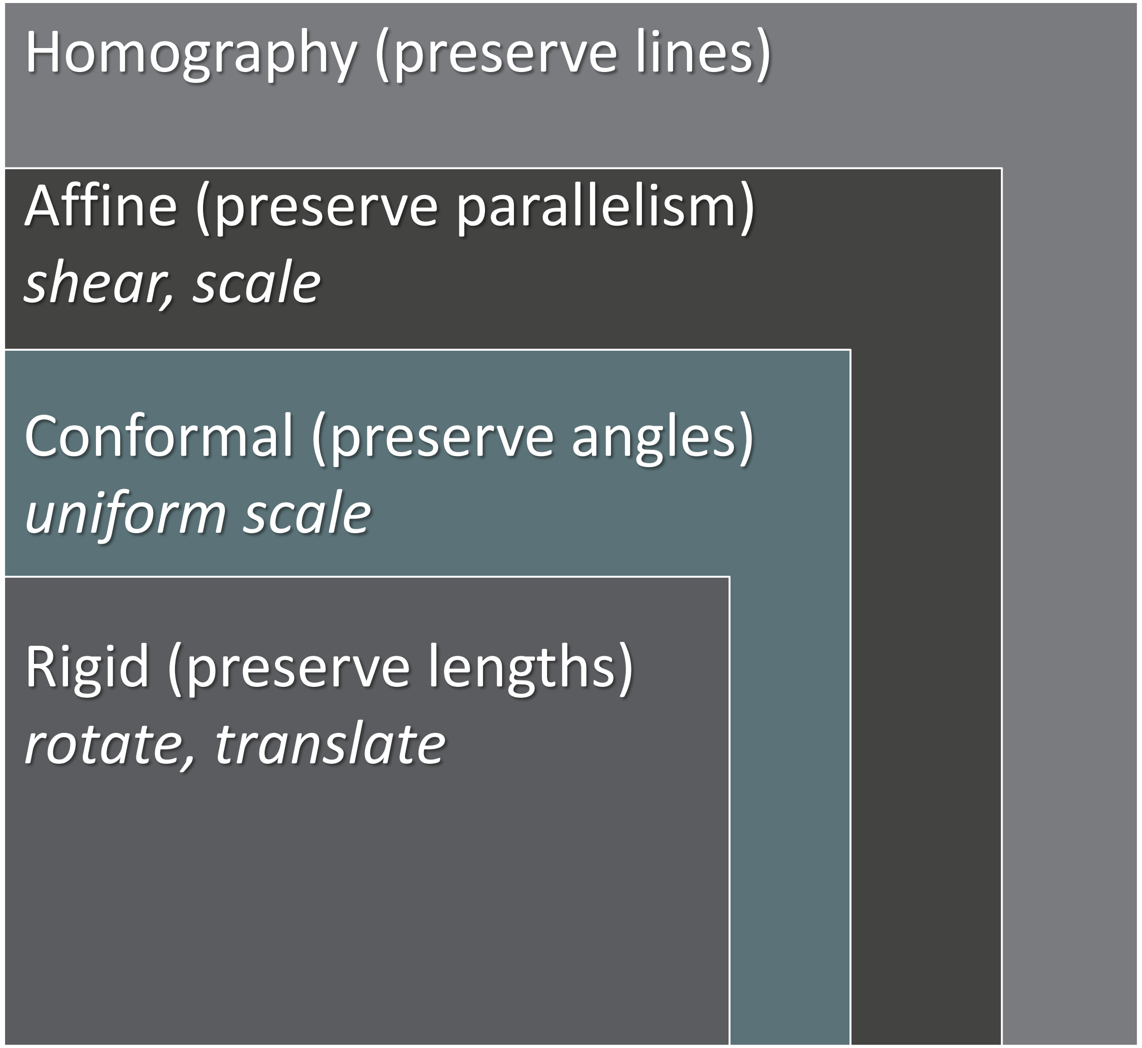
…a more restricted set of transformations also preserve parallelism in lines. These are called **Affine** transforms.

…transforms that further preserve the angle between lines are called **Conformal**.

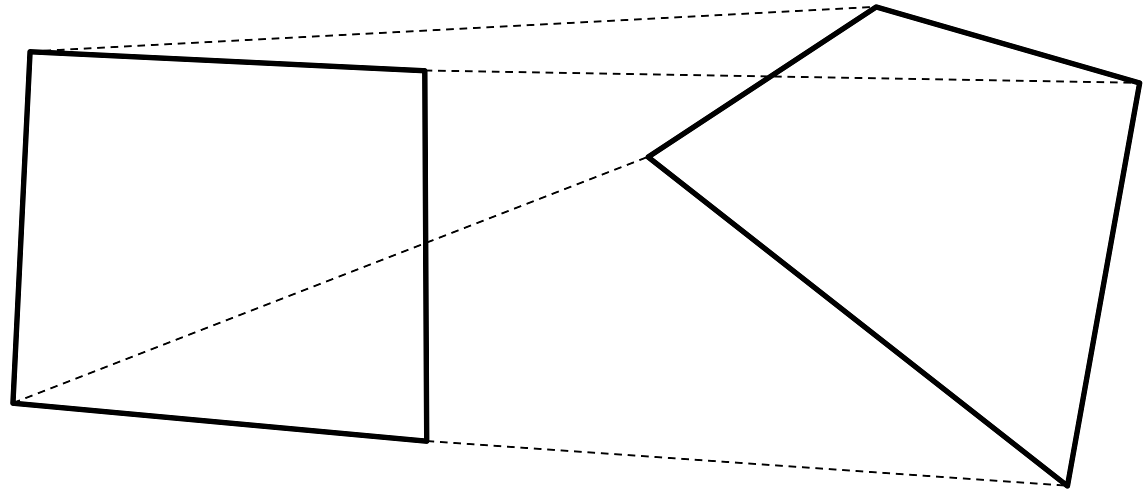
…transforms that additionally preserve the lengths of line segments are called **Rigid**.

Where do translate, rotate and scale fit into these?

Properties of 2D transforms



Homography: mapping four points



How does the mapping of 4 points uniquely define the 3x3 Homography matrix?

Affine: preserving parallel lines

What restriction does the Affine property impose on H?

If two lines are parallel their intersection point at infinity, is of the form $[x\ y\ 0]^T$.

If these lines map to lines that are still parallel, then $[x\ y\ 0]^T$ transformed must continue to map to a point at infinity or $[x'\ y'\ 0]^T$

i.e. $$[x'\ y'\ 0]^T = \begin{pmatrix} \begin{pmatrix} & A & \\ & & \end{pmatrix} & \begin{pmatrix} t \\ \ \end{pmatrix} \\ 0\quad 0 & 1 \end{pmatrix} [x\ y\ 0]^T$$

In Cartesian co-ordinates Affine transforms can be written as:

$$p' = Ap + t$$

Affine properties: composition

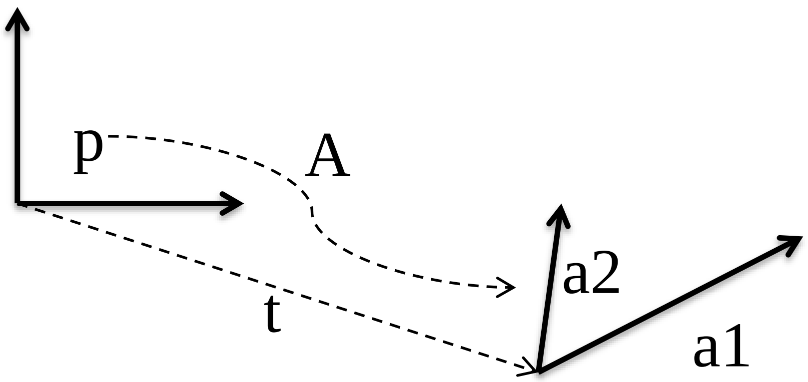
Affine transforms are closed under composition. i.e.
Applying transform (A_1,t_1) (A_2,t_2) in sequence results in an overall Affine transform.

$$p' = A_2 (A_1 p + t_1) + t_2 \quad \Rightarrow (A_2 A_1)p + (A_2 t_1 + t_2)$$

Inverse?

Affine transform: geometric interpretation

A change of basis vectors and translation of the origin



point p in the local coordinates of a reference frame defined by <a1,a2,t> is

$$\begin{pmatrix} \begin{pmatrix} a1 & a2 \end{pmatrix} & \begin{pmatrix} t \end{pmatrix} \\ 0 \quad 0 & 1 \end{pmatrix}^{-1} \begin{pmatrix} p \end{pmatrix}$$

Composing Transformations

Any sequence of linear transforms can be collapsed into a single 3x3 matrix by concatenating the transforms in the sequence.

In general transforms DO NOT commute, however certain combinations of transformations are commutative…

try out various combinations of translate, rotate, scale.

Rotation about a fixed point

The typical rotation matrix, rotates points about the origin.
To rotate about specific point *q*, use the ability to compose transforms…

$$T_q \; R \; T_{-q}$$

Representing 3D transforms as a 4x4 matrix

Translate a point $[x\ y\ z]^T$ by $[t_x\ t_y\ t_z]^T$:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

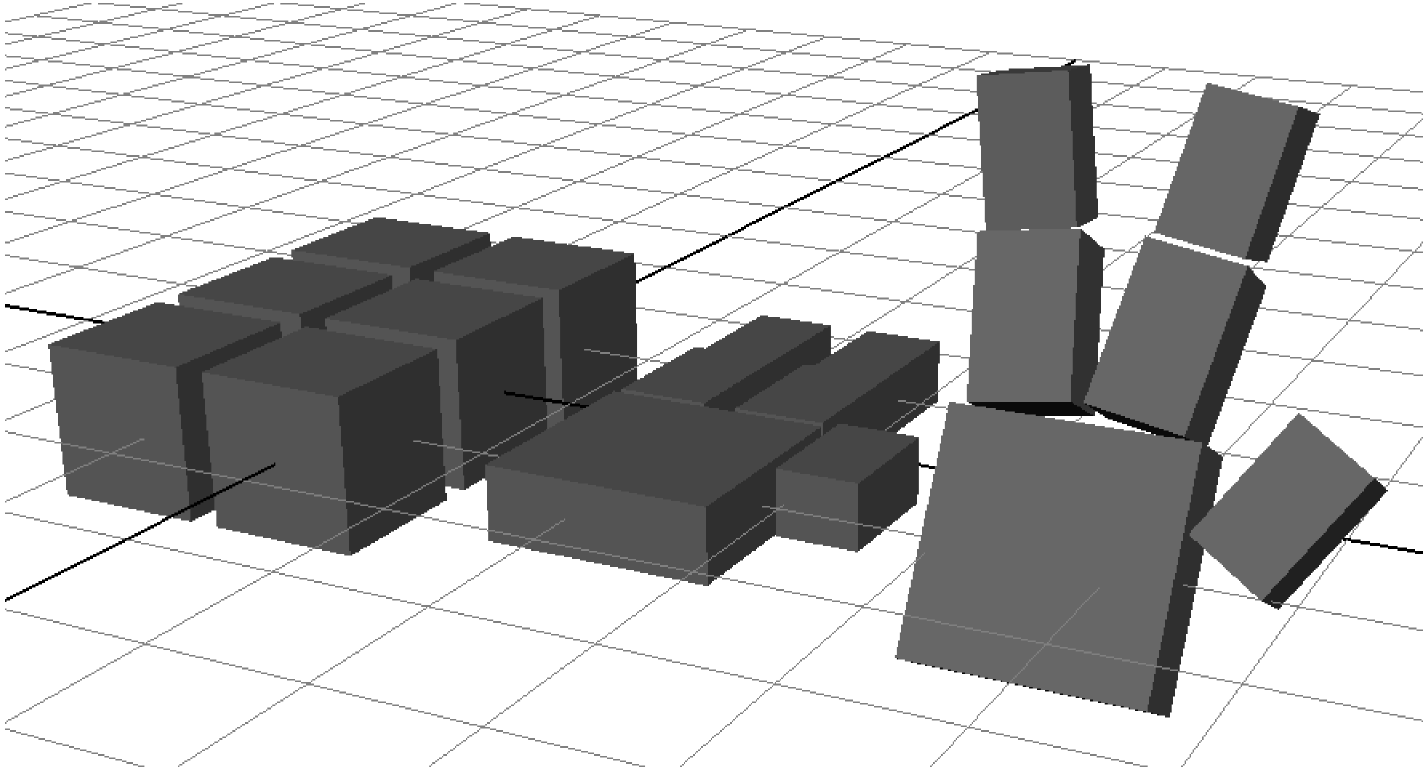
Rotate a point $[x\ y\ z]^T$ by an angle t around z axis:

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos t & -\sin t & 0 & 0 \\ \sin t & \cos t & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

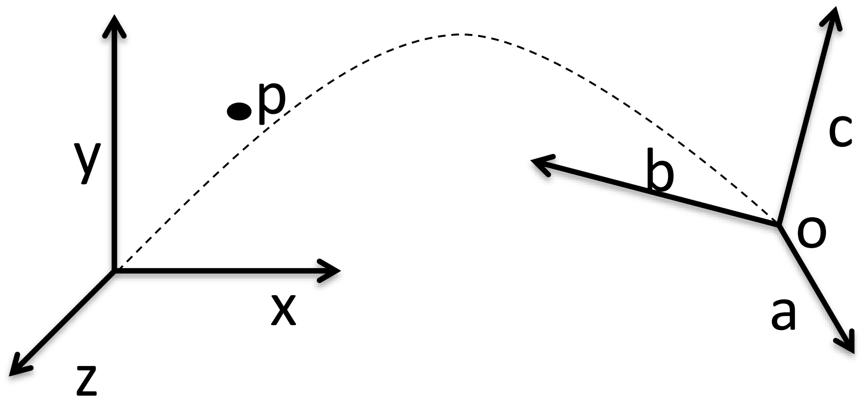
Scale a point $[x\ y\ z]^T$ by a factor $[s_x\ s_y\ s_z]^T$

$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Scene Hierarchies



Change of reference frame/basis matrix



$$p = ap_x' + bp_y' + cp_z' + o$$

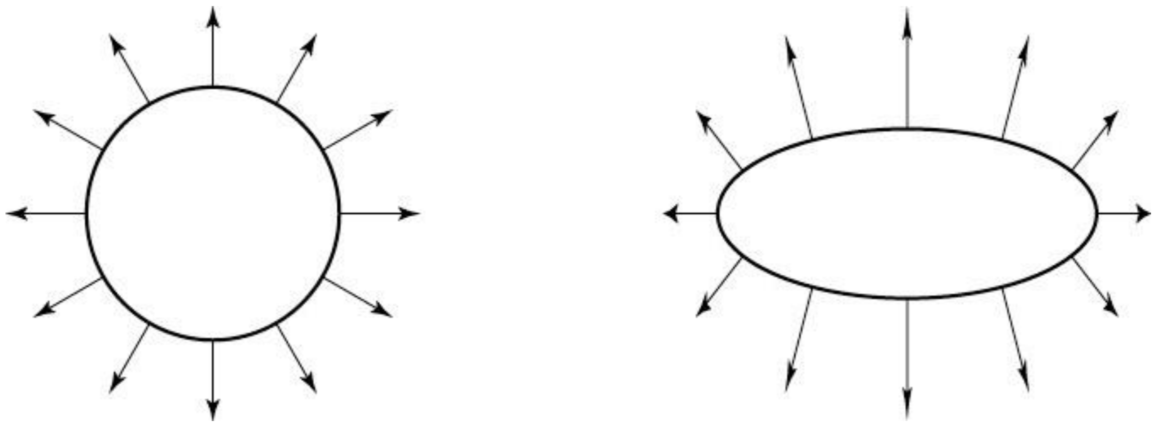
$$p = \begin{pmatrix} a\ b\ c\ o \\ 0\ 0\ 0\ 1 \end{pmatrix} p'$$

$$p' = \begin{pmatrix} a\ b\ c\ o \\ 0\ 0\ 0\ 1 \end{pmatrix}^{-1} p$$

Transforming normal vectors

Transforming surface normals

- differences of points (and therefore tangents) transform OK
- normals do not --> use inverse transpose matrix

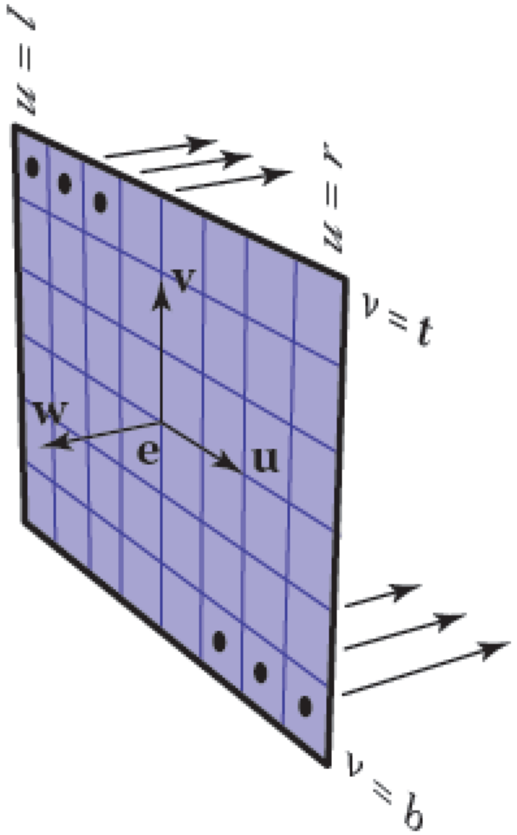


$$\begin{aligned}
&\text{have: } \mathbf{t} \cdot \mathbf{n} = \mathbf{t}^T \mathbf{n} = 0 \\
&\text{want: } M\mathbf{t} \cdot X\mathbf{n} = \mathbf{t}^T M^T X \mathbf{n} = 0 \\
&\text{so set } X = (M^T)^{-1} \\
&\text{then: } M\mathbf{t} \cdot X\mathbf{n} = \mathbf{t}^T M^T (M^T)^{-1} \mathbf{n} = \mathbf{t}^T \mathbf{n} = 0
\end{aligned}$$

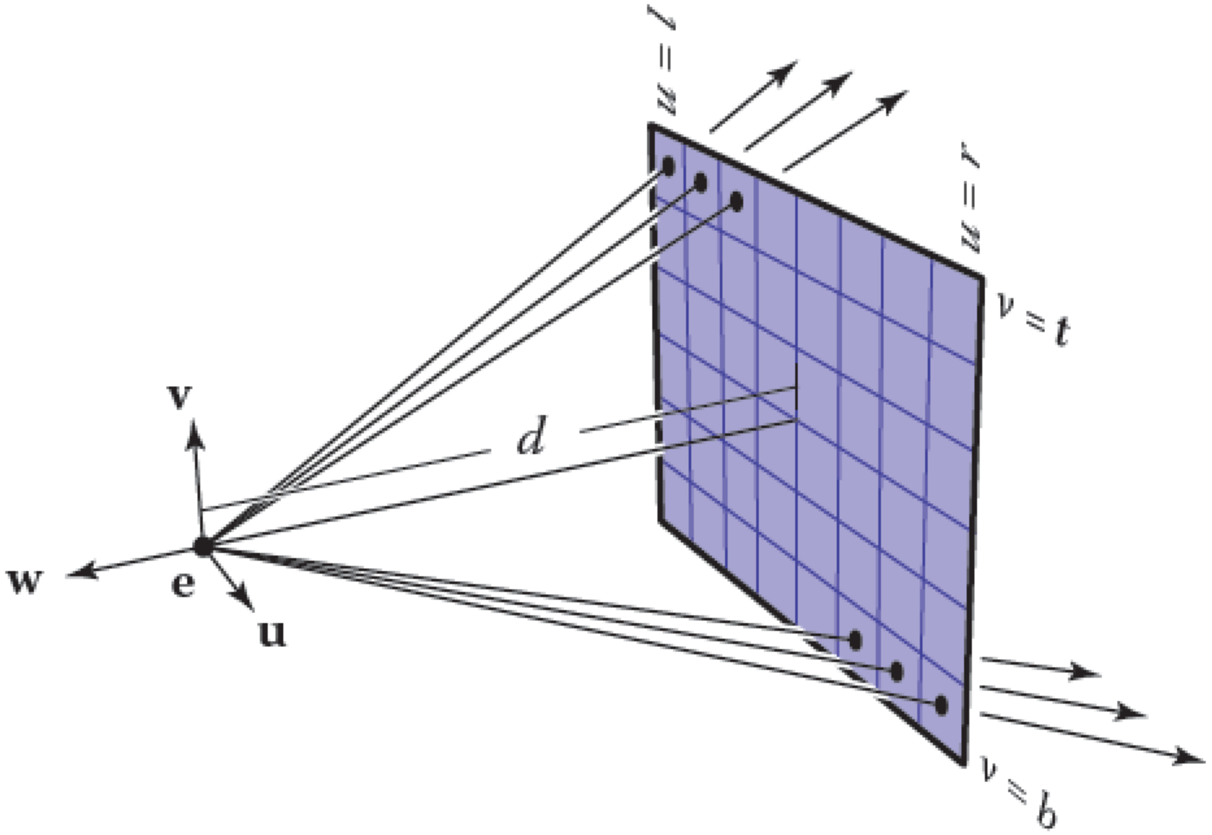
Topic 9.

Viewing and Projection

Reminder: Camera Model

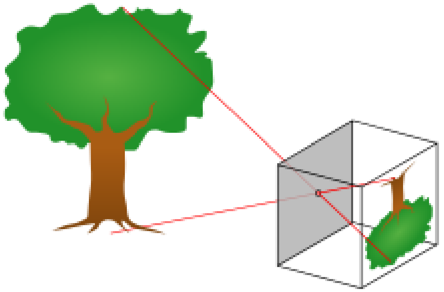


Parallel projection
same direction, different origins



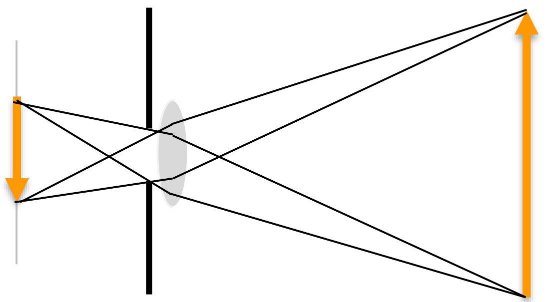
Perspective projection
same origin, different directions

Camera model: camera obscura

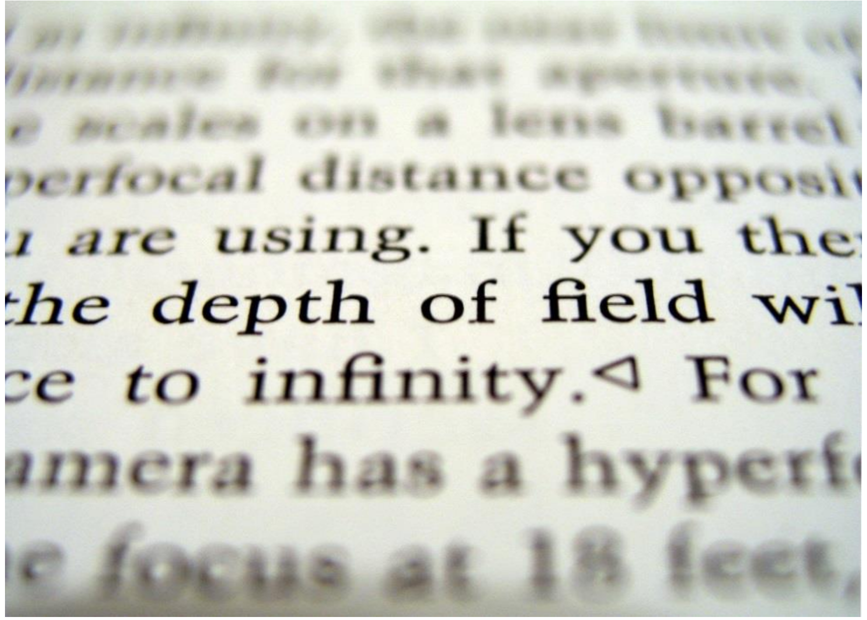


Camera model

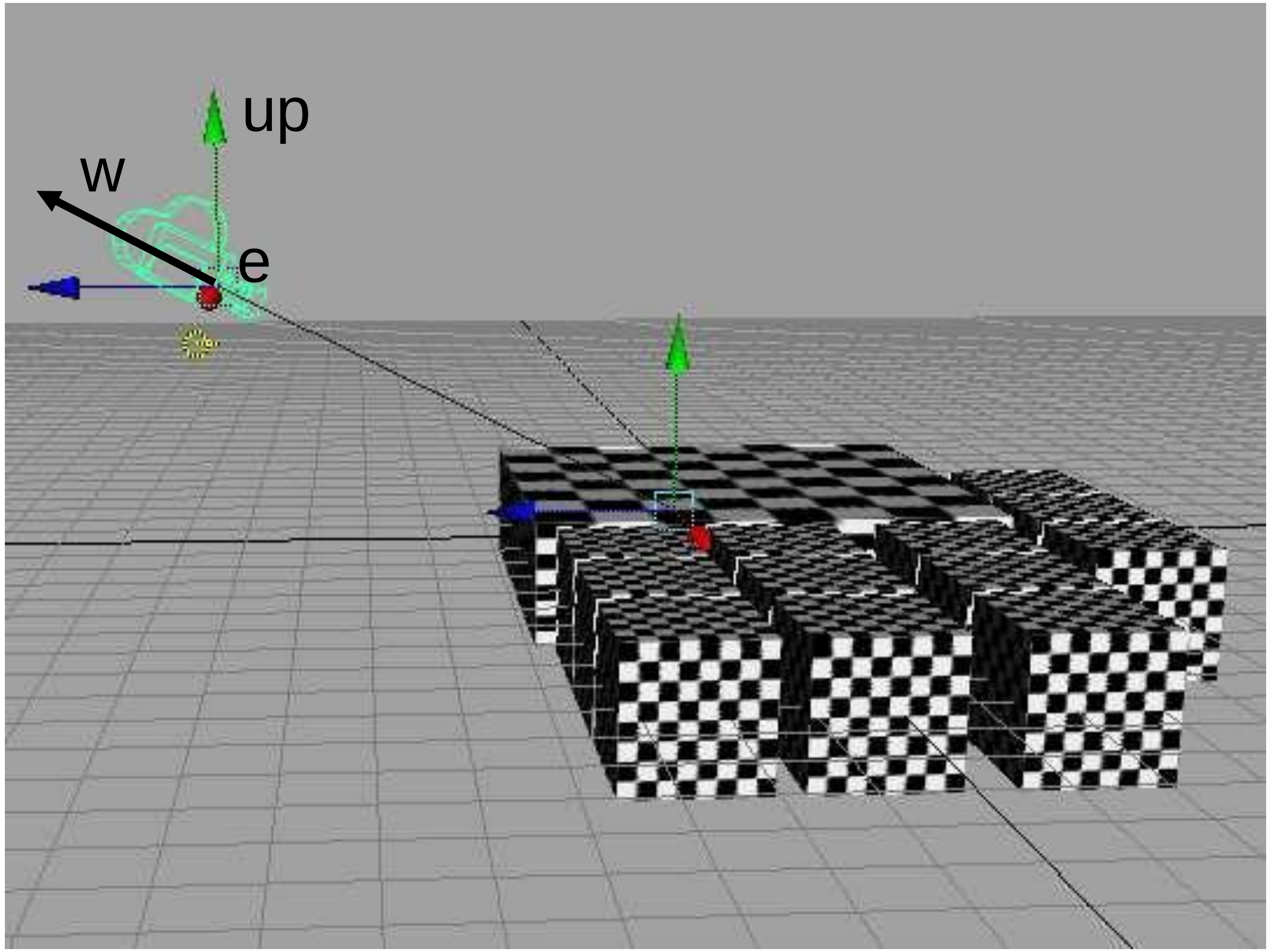
Camera with a lens



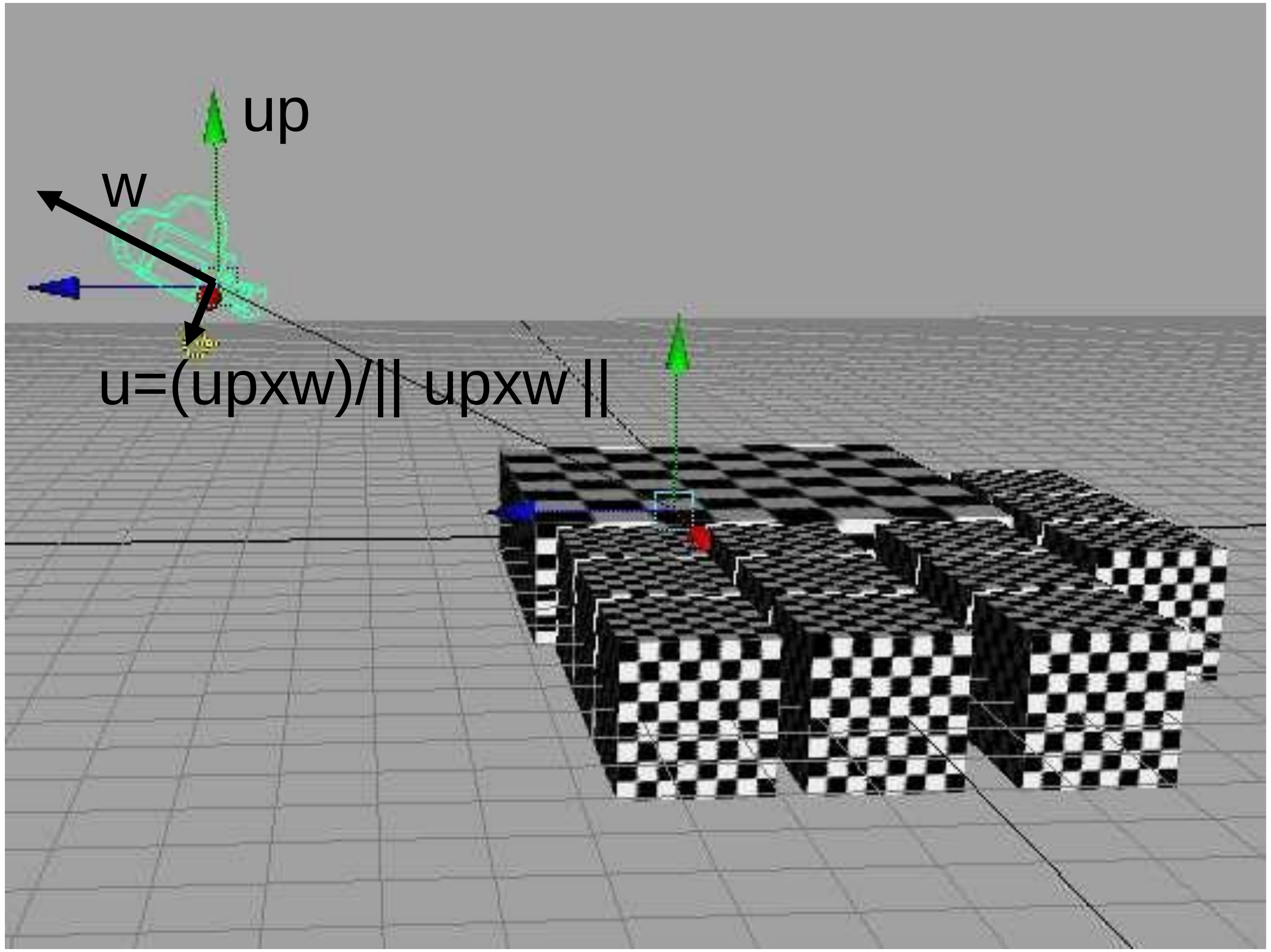
Depth of Field



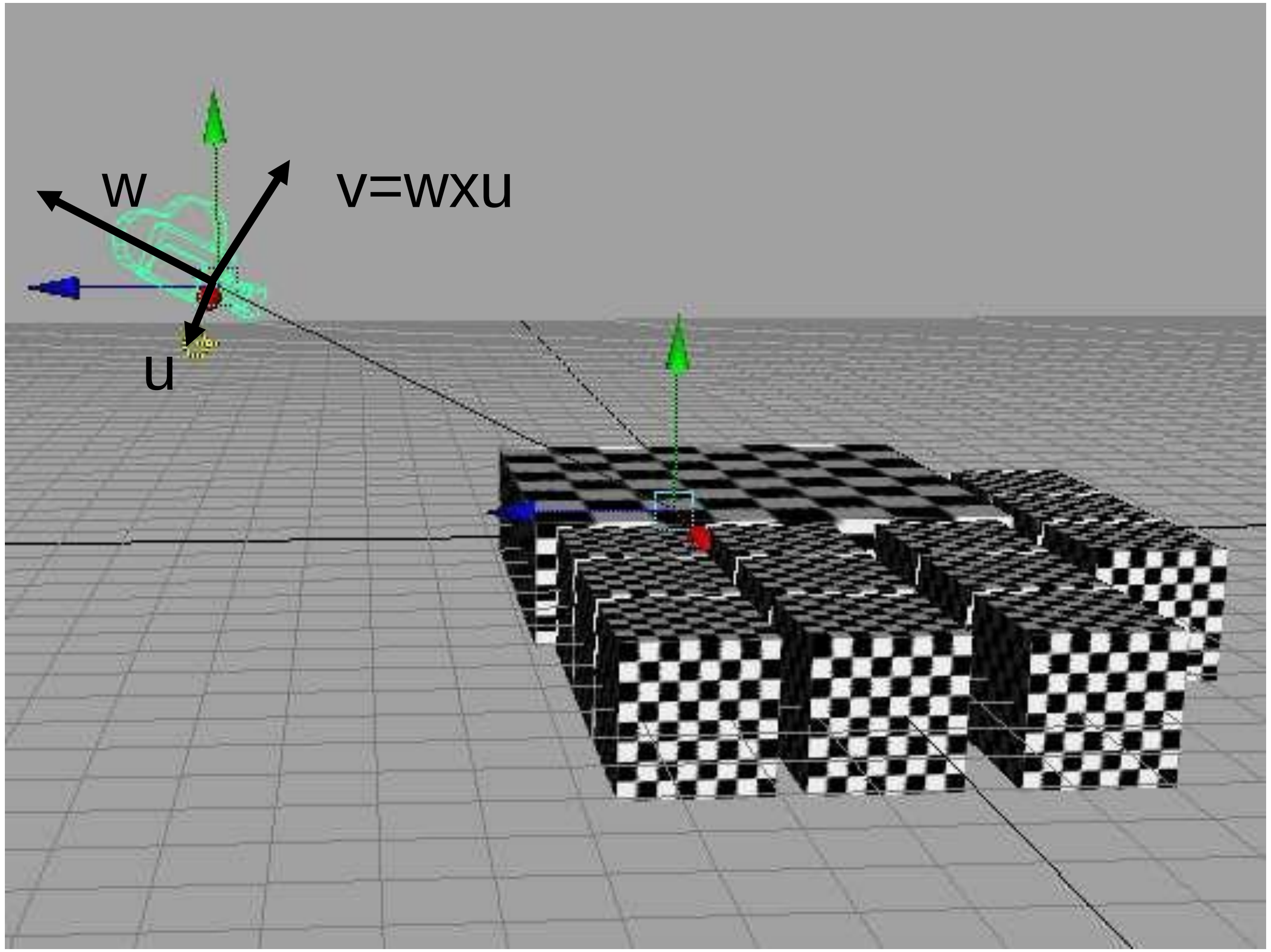
Viewing Transform



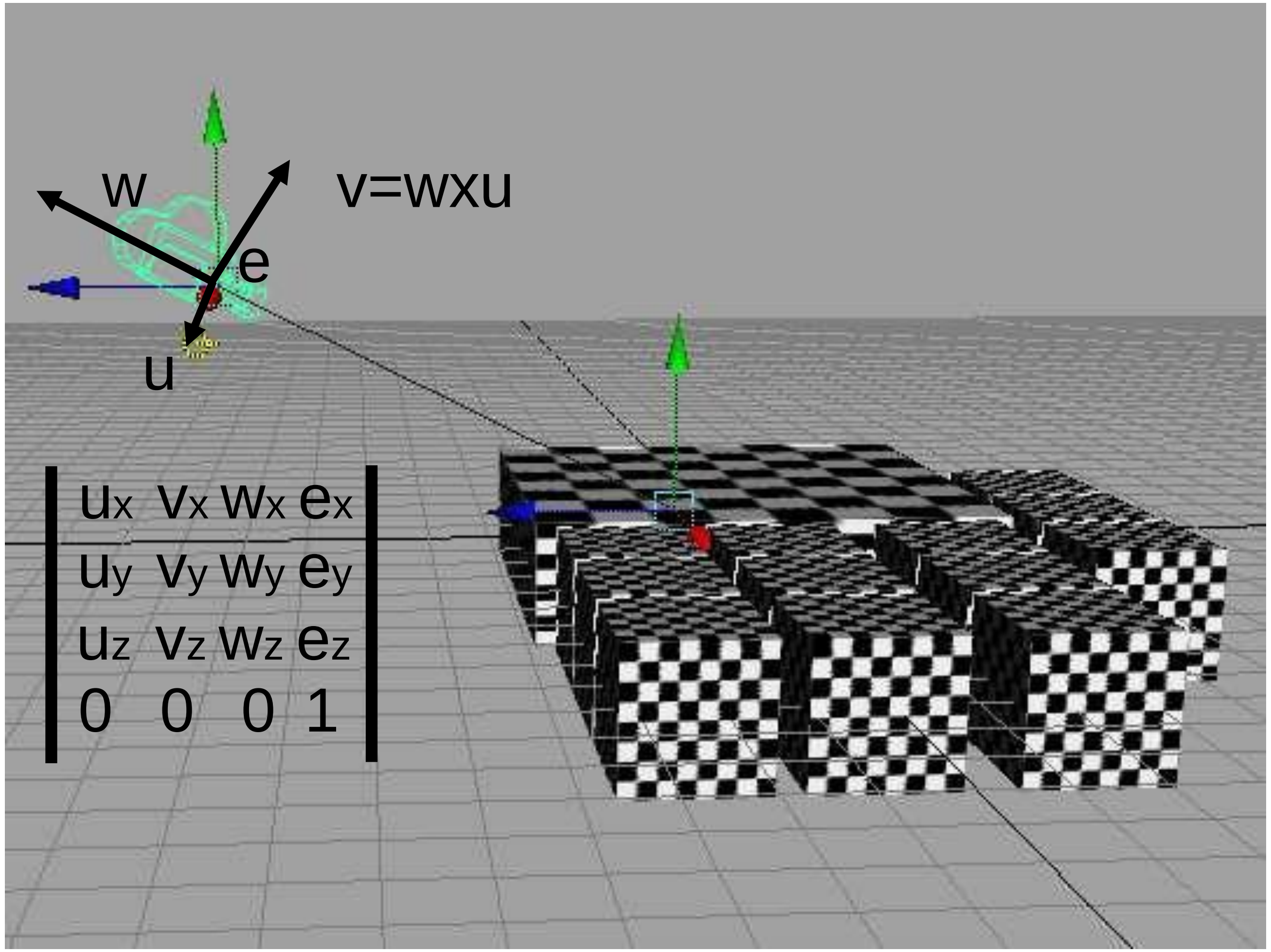
Viewing Transform



Viewing Transform



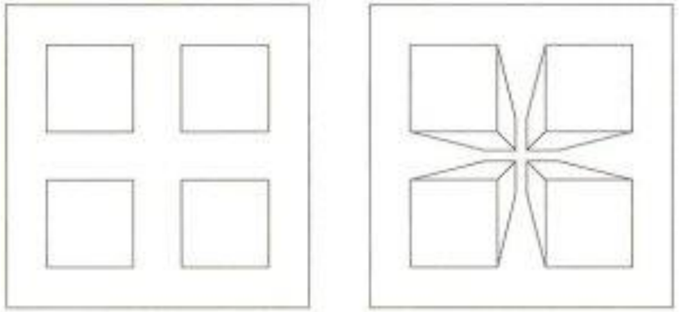
Change-of-basis Matrix



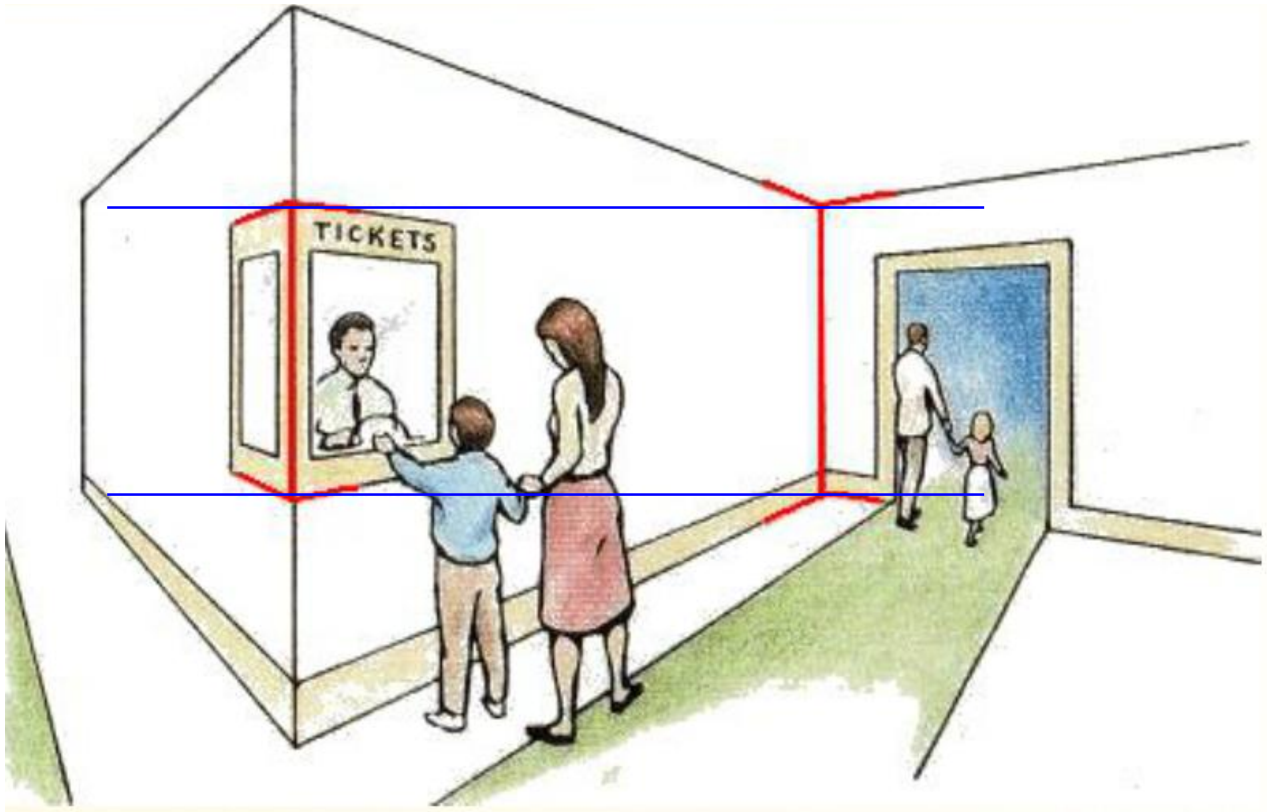
$$\begin{bmatrix} u_x & v_x & w_x & e_x \\ u_y & v_y & w_y & e_y \\ u_z & v_z & w_z & e_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

Camera model

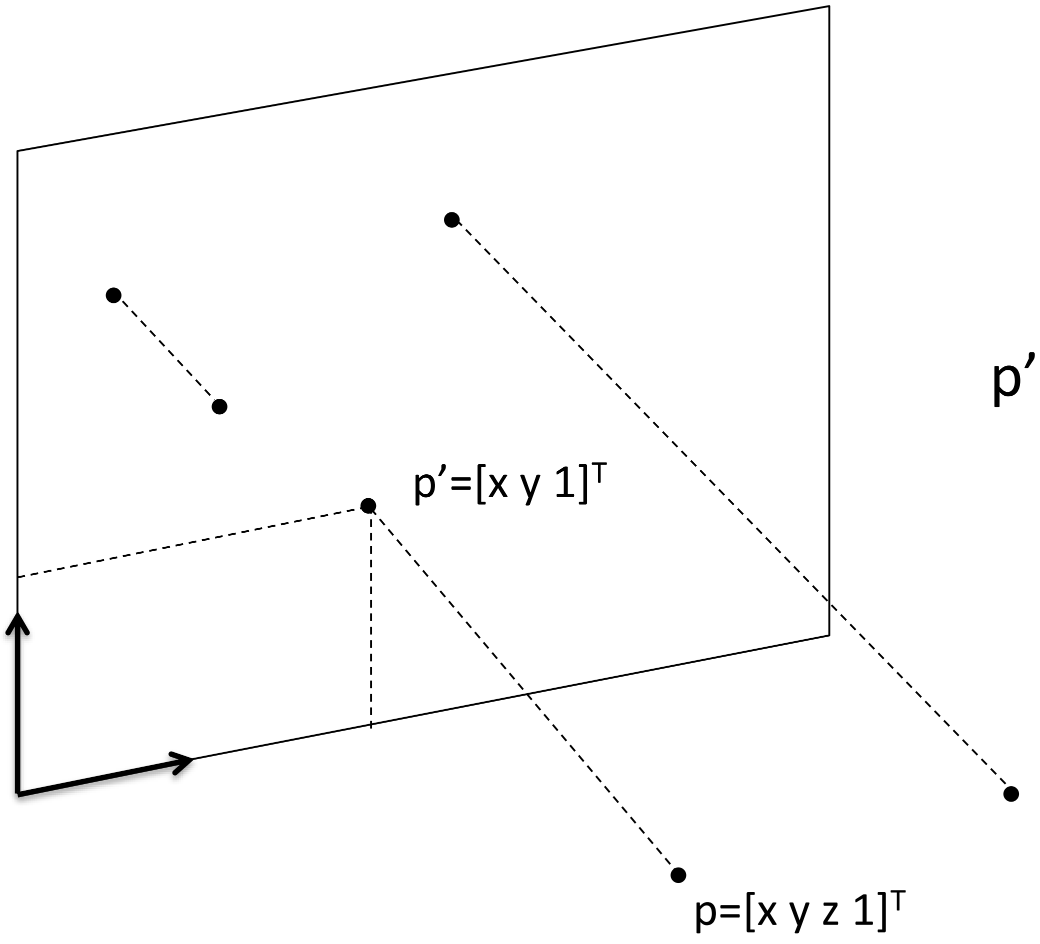
What is the difference between these images?



Perspective: Muller-Lyer Illusion

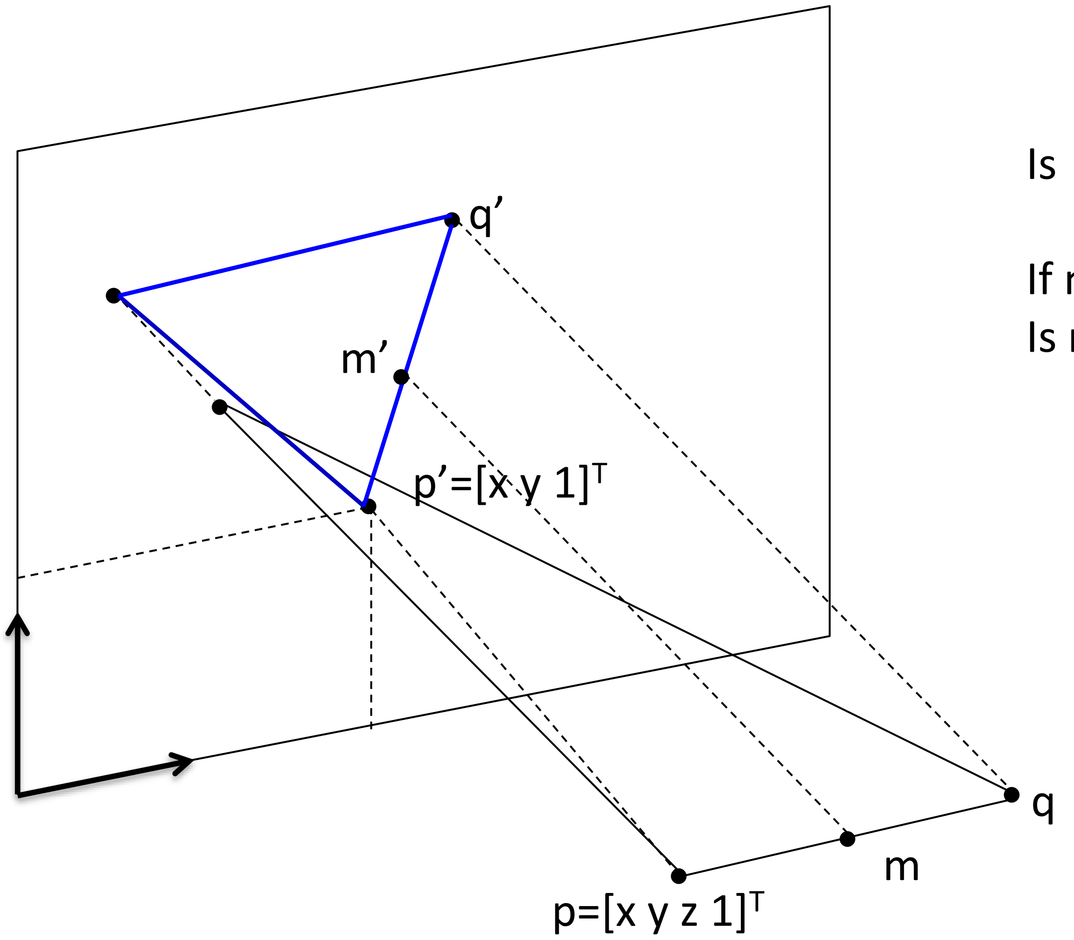


Orthographic projection



$$p' = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} p$$

Orthographic projection

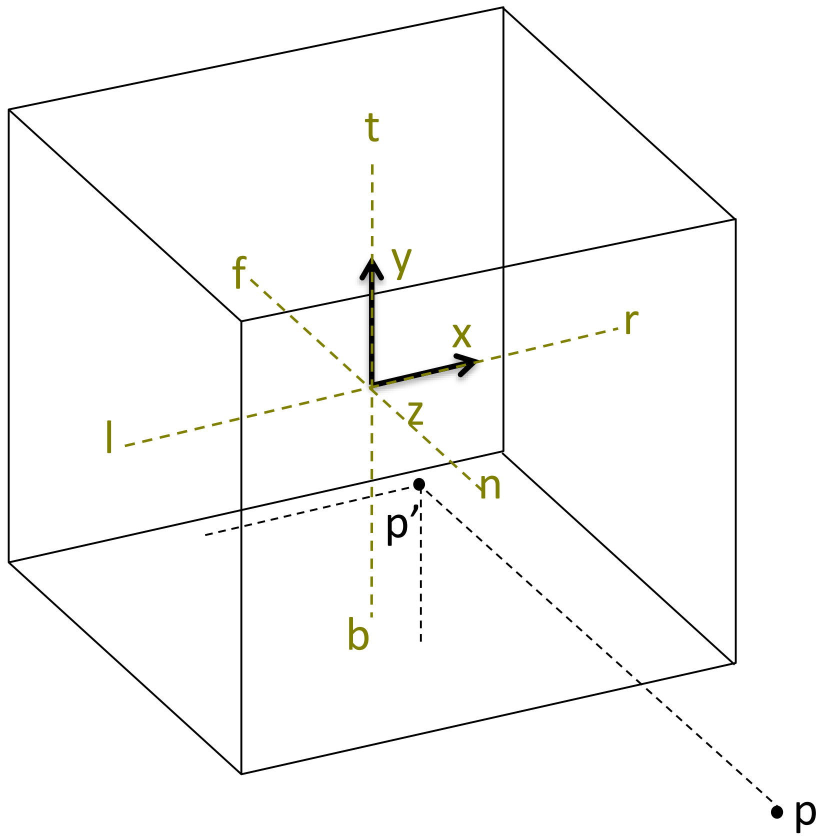


no: since z-coordiinate…

Is |p-q| = |p'-q'| ?

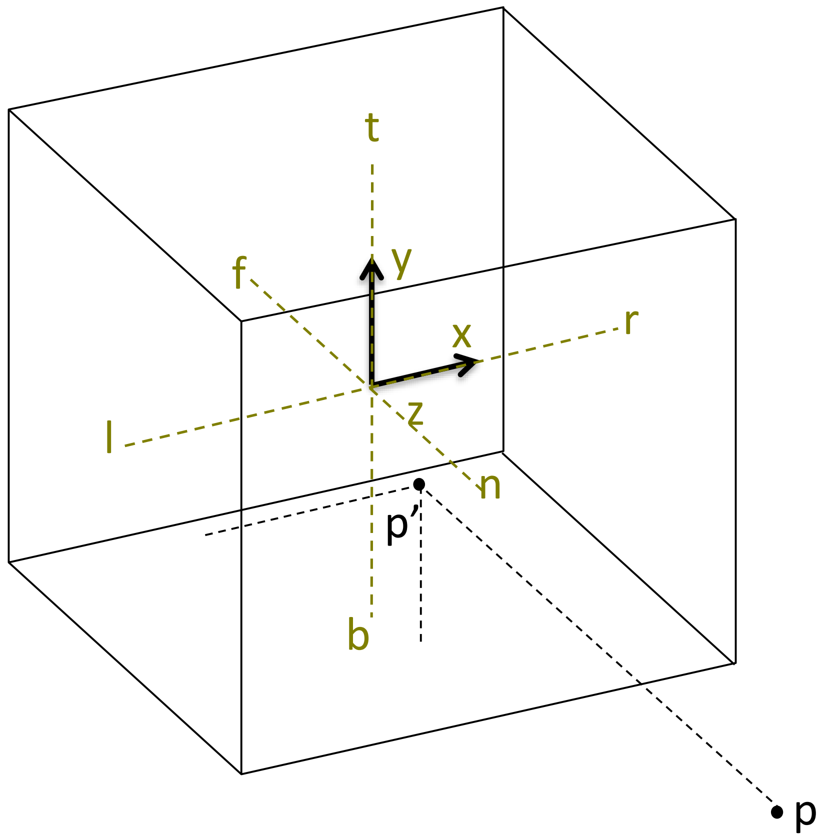
If m= (p+q)/2,
Is m' = (p'+q')/2?

Cannonical view volume



Map 3D to a cube centered at the origin of side length 2!

Cannonical view volume

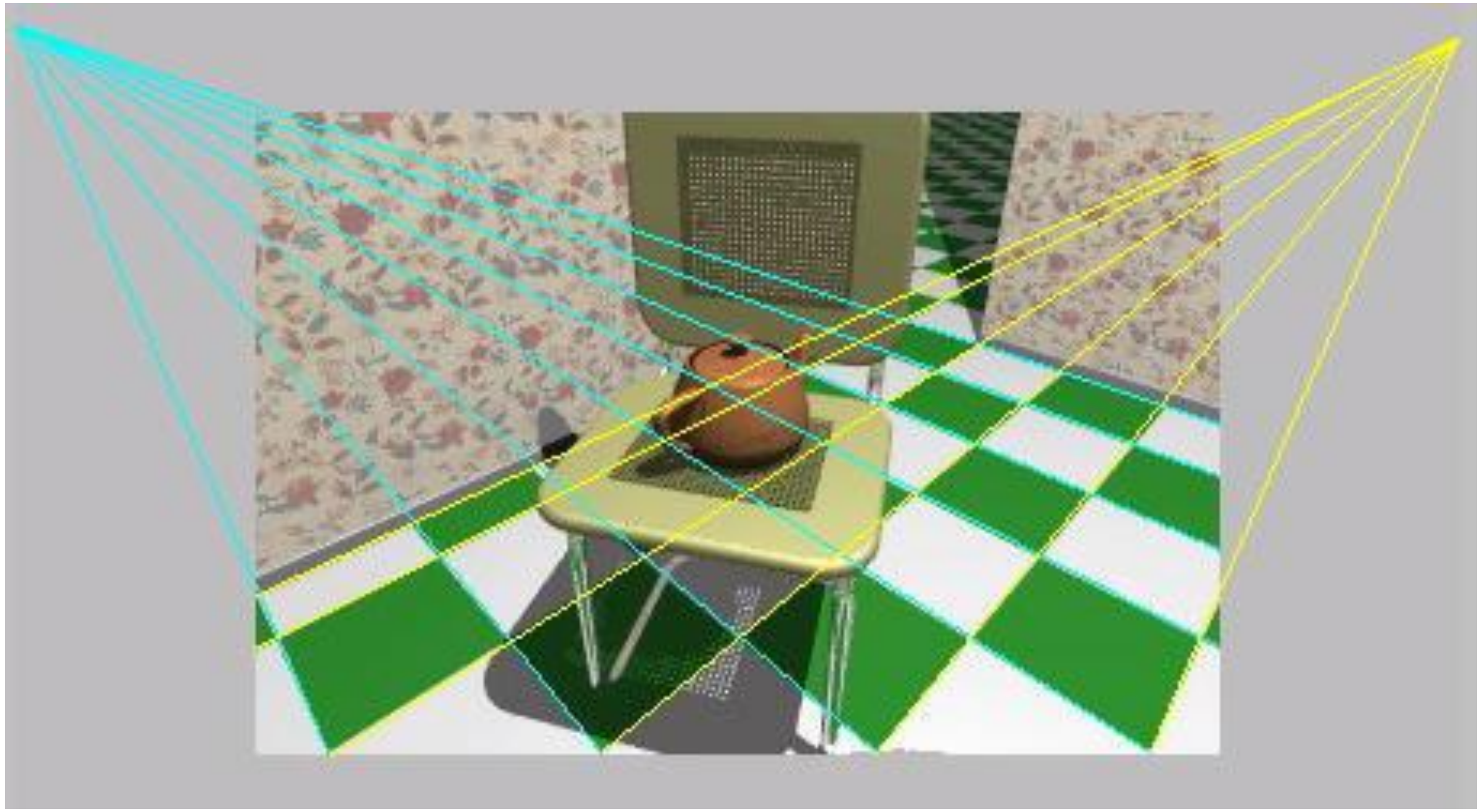


Map 3D to a cube centered
at the origin of side length 2!

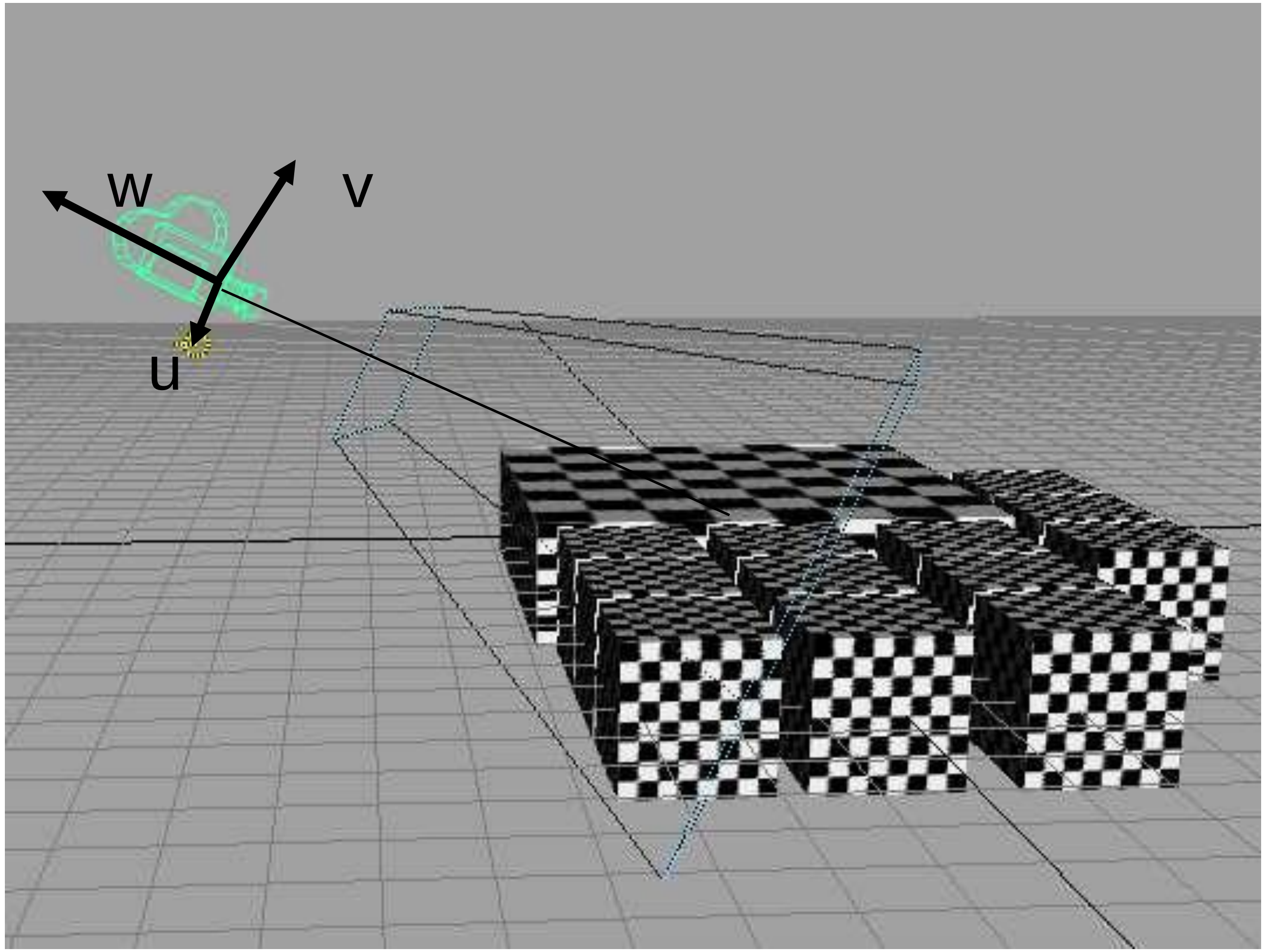
Translate(-(l+r)/2,-(t+b)/2,-(n+f)/2))
Scale(2/(r-l), 2/(t-b), 2/(f-n))

Camera model

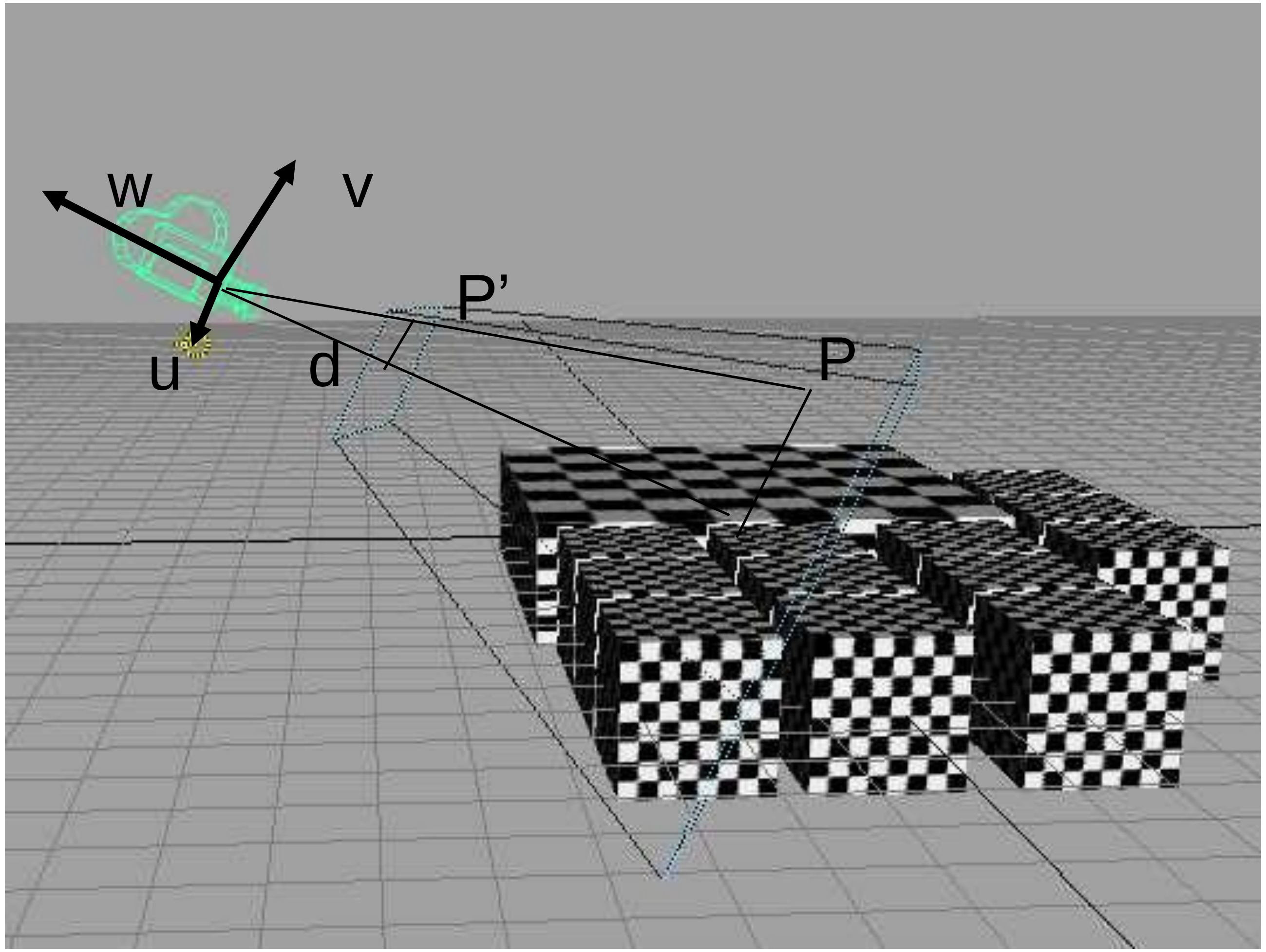
Perspective Projection



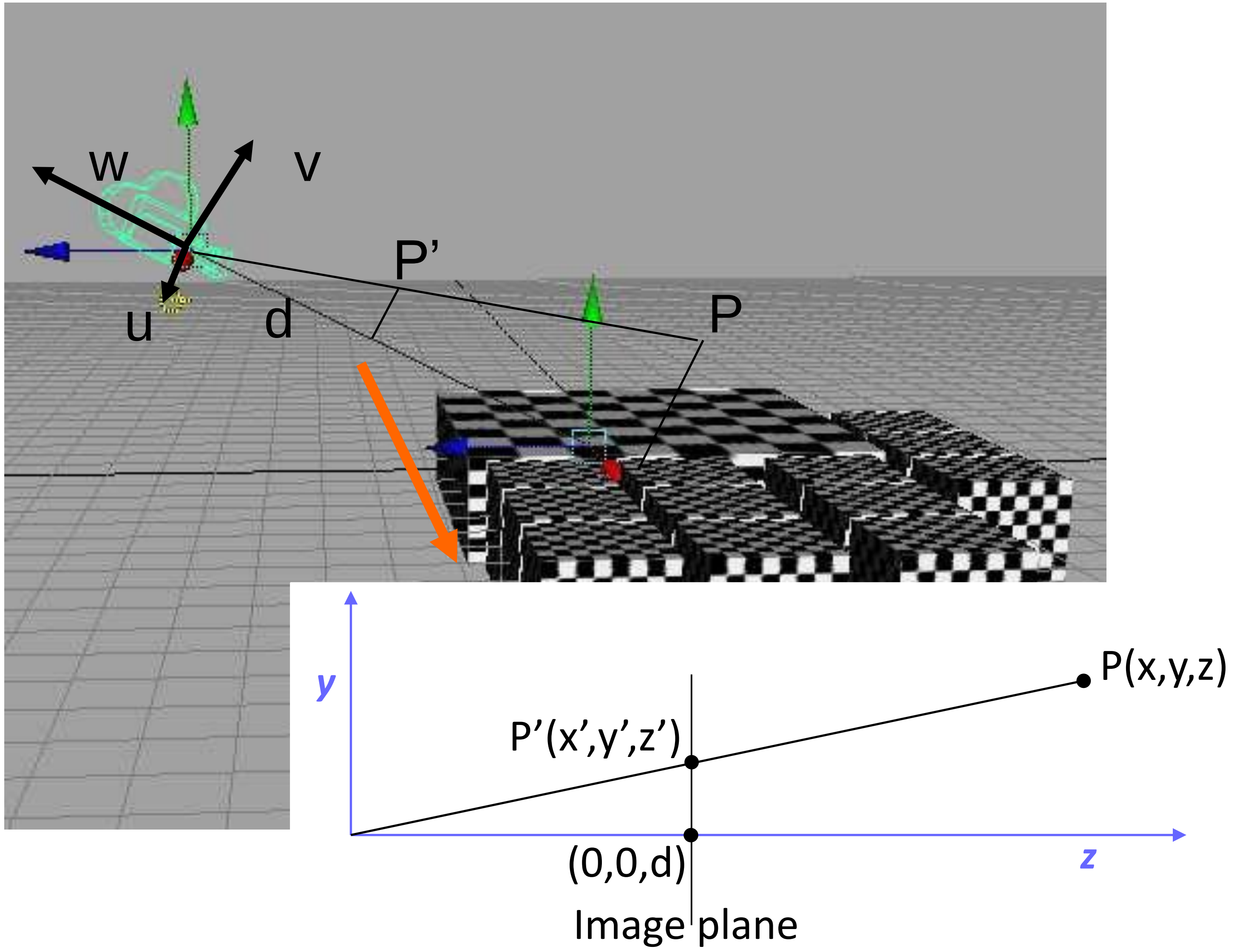
Perspective projection



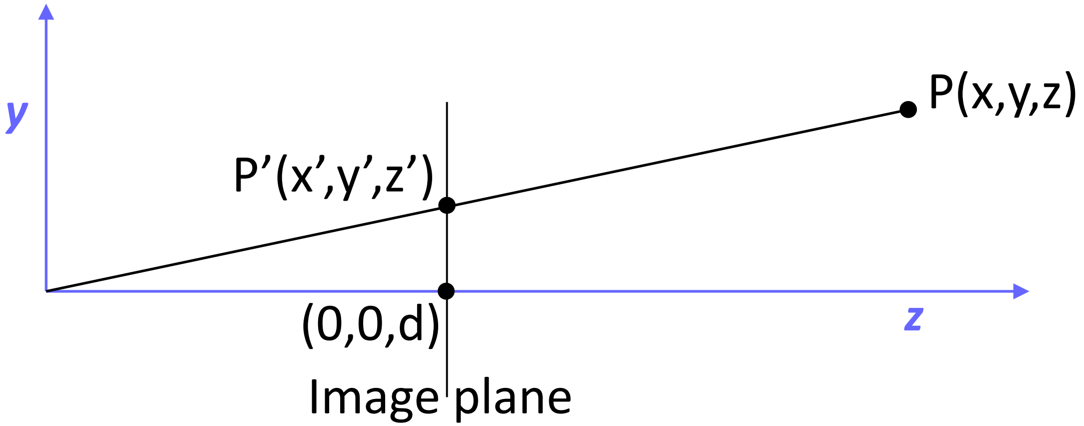
Perspective projection



Simple Perspective



Simple Perspective

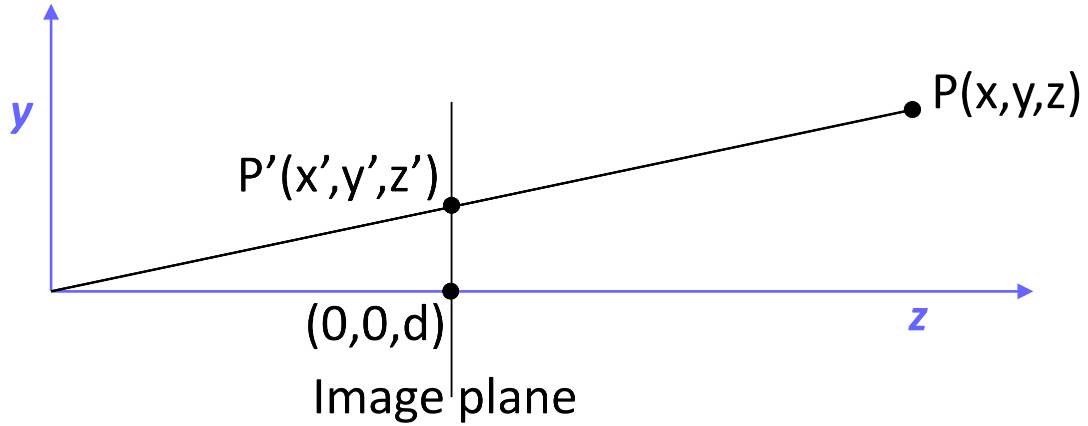


$y' = yd/z$

$x' = xd/z$

$z' = d$

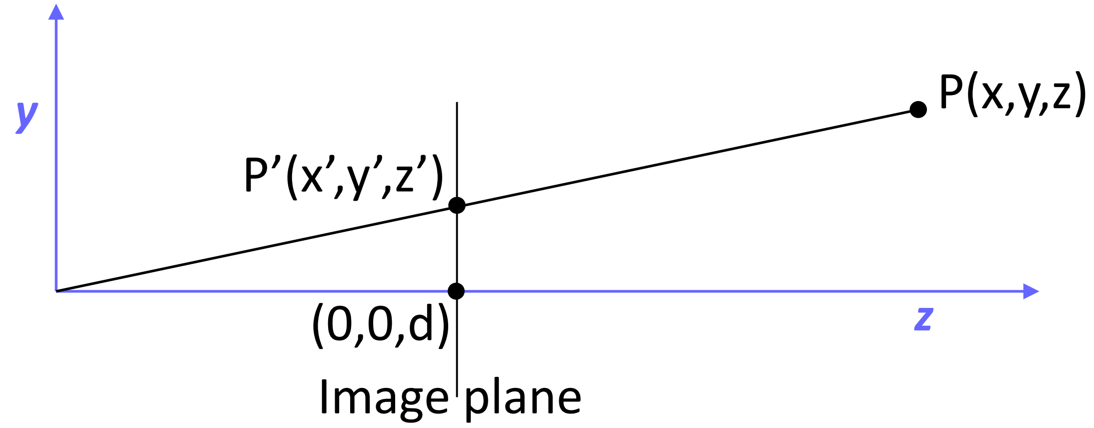
Simple Perspective



$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$w' = z/d$

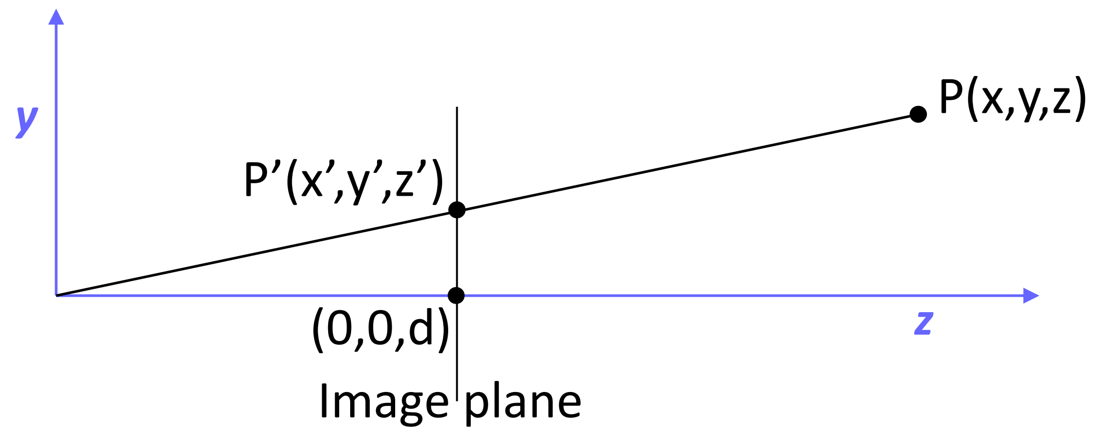
Simple Perspective



$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & a & b \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

Find **a** and **b** such that z'=-1 when z=d and z'=1 when z=D, where d and D are near and far clip planes.

Simple Perspective

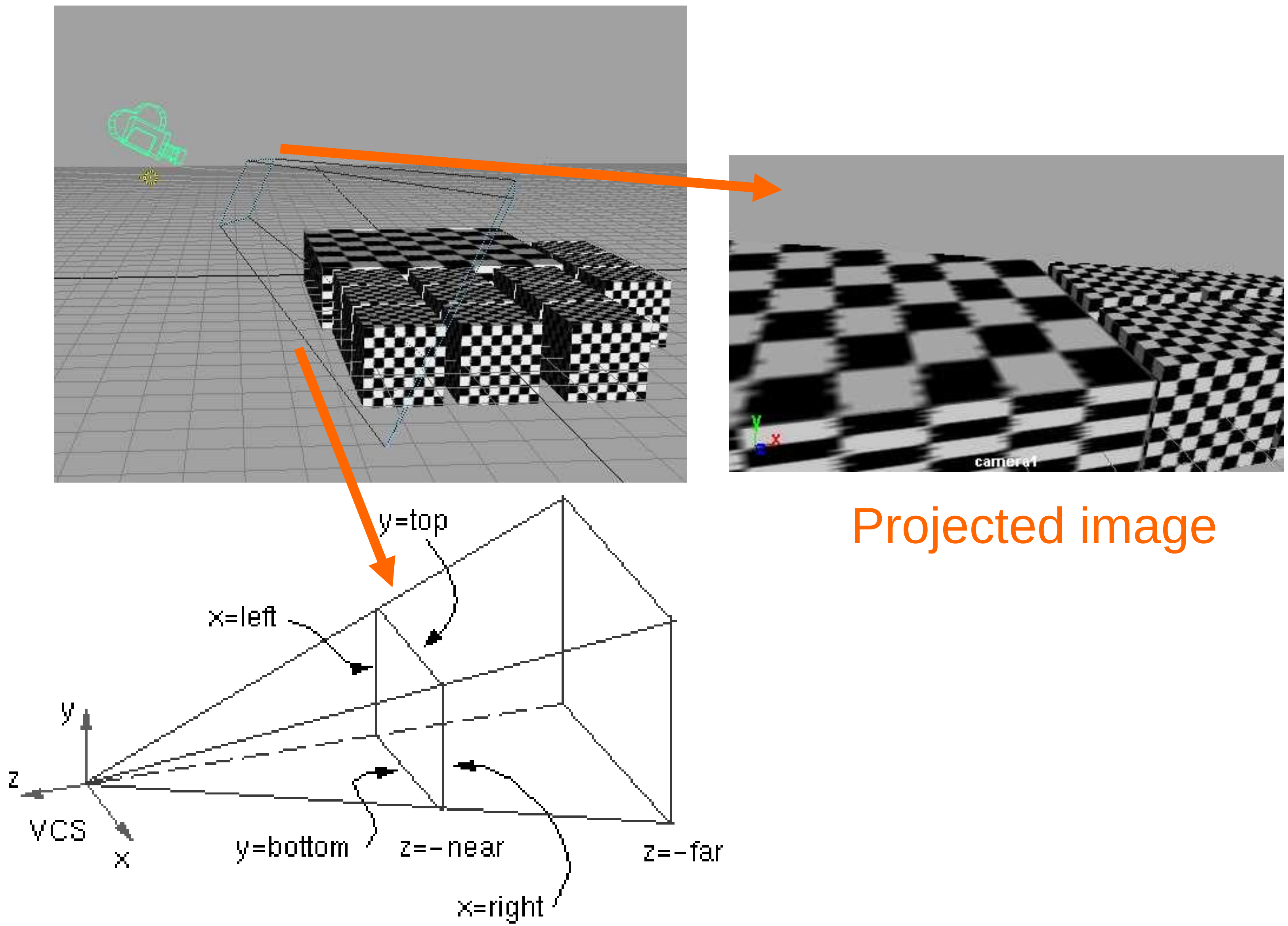


$$\begin{pmatrix} x' \\ y' \\ z' \\ w' \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \color{red}{a} & \color{red}{b} \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

$z'=d(\mathbf{a}z+\mathbf{b})/z \Rightarrow -1=\mathbf{a}d+\mathbf{b}$ and $1=d(\mathbf{a}D+\mathbf{b})/D$

=> $\mathbf{b}=2D/(d-D)$ and $\mathbf{a}=(D+d)/(d(D-d))$

Viewing volumes



Projected image

Viewing Pipeline