CS130 : Computer Graphics

Lecture 8: Viewing Transformations

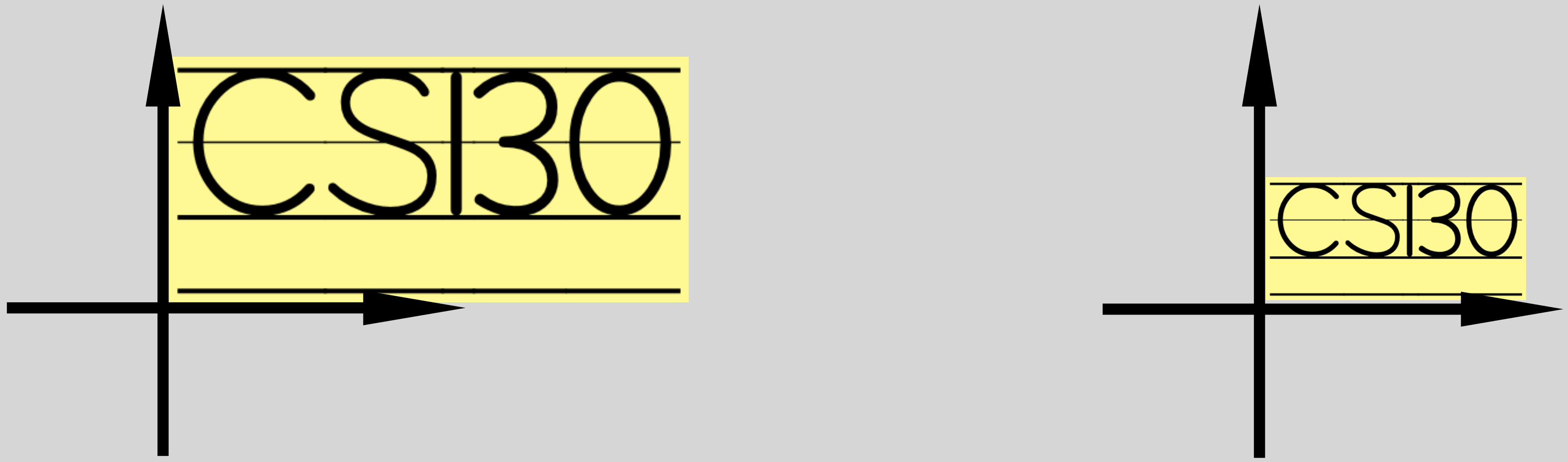
Tamar Shinar
Computer Science & Engineering
UC Riverside

2D Transformations

Uniform Scale

$$\begin{pmatrix} s & 0 \\ 0 & s \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} sx \\ sy \end{pmatrix}$$

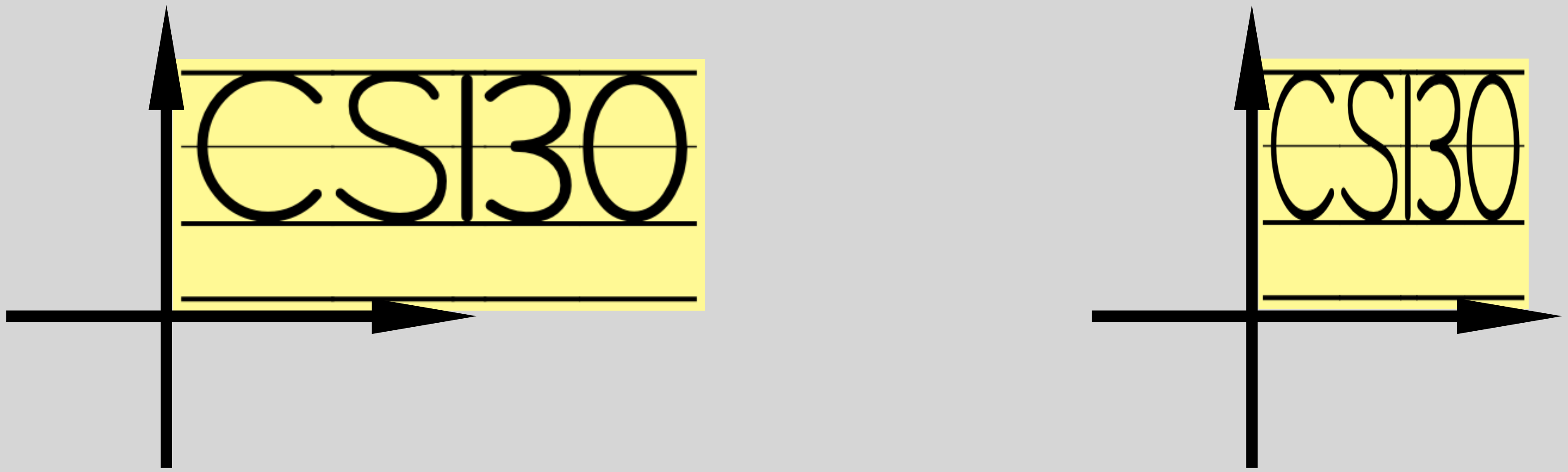
$$\begin{pmatrix} .5 & 0 \\ 0 & .5 \end{pmatrix}$$



Nonuniform Scale

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

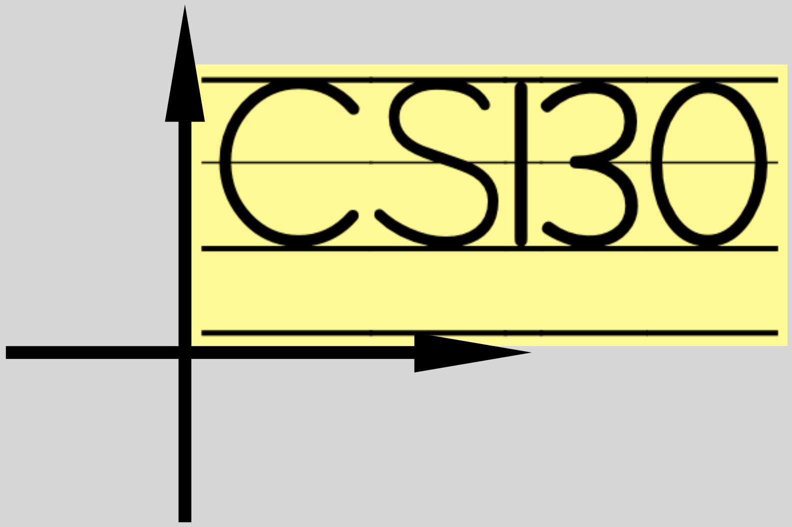
$$\begin{pmatrix} .5 & 0 \\ 0 & 1 \end{pmatrix}$$

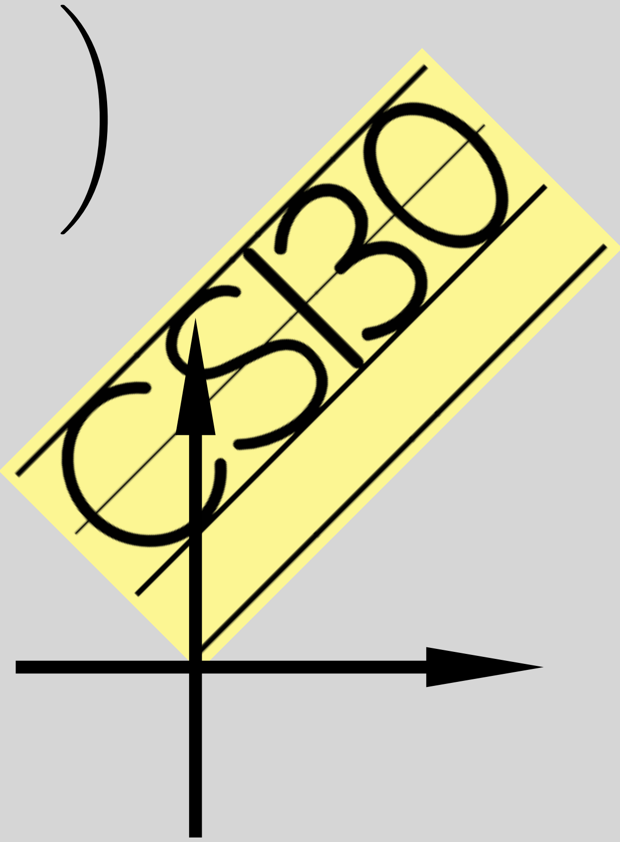


Rotation

$$\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x\cos\theta - y\sin\theta \\ x\sin\theta + y\cos\theta \end{pmatrix}$$

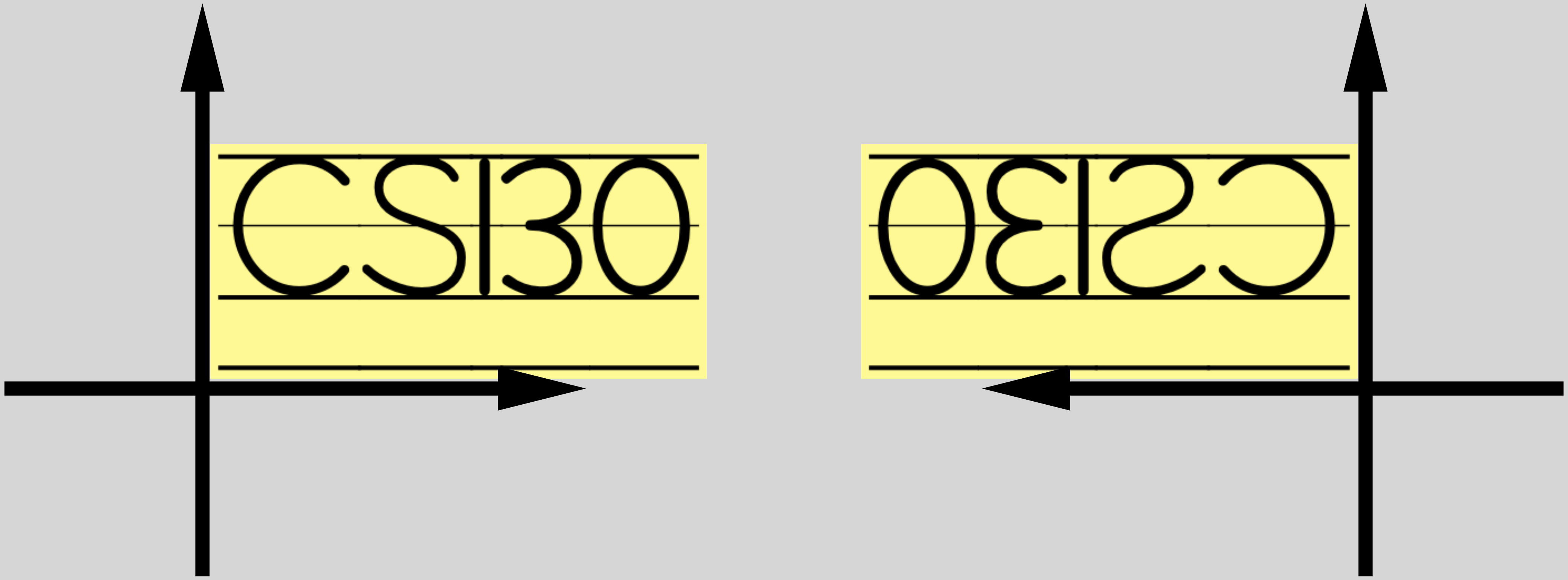
$$\begin{pmatrix} \cos\frac{\pi}{4} & -\sin\frac{\pi}{4} \\ \sin\frac{\pi}{4} & \cos\frac{\pi}{4} \end{pmatrix}$$





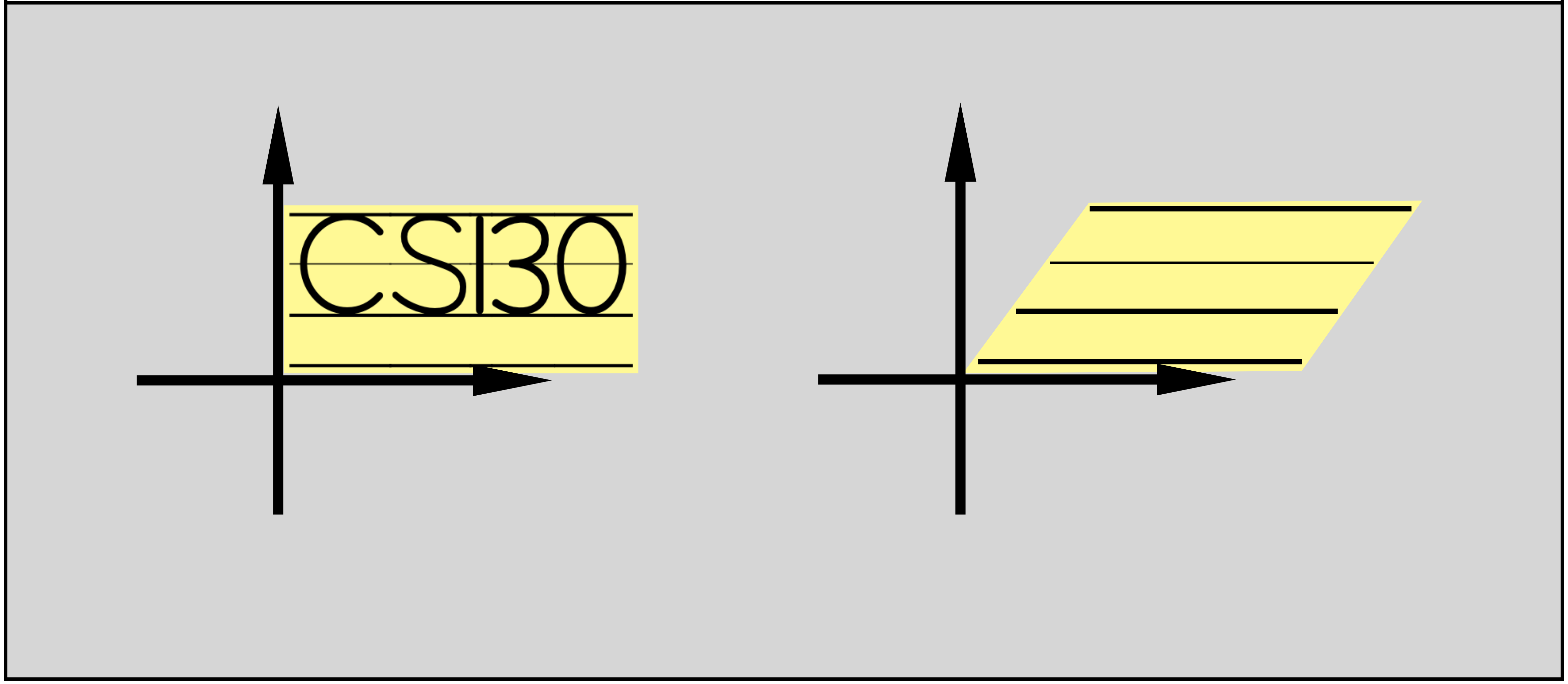
Reflection

$$\begin{pmatrix} -1 & 0 \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} -x \\ y \end{pmatrix}$$



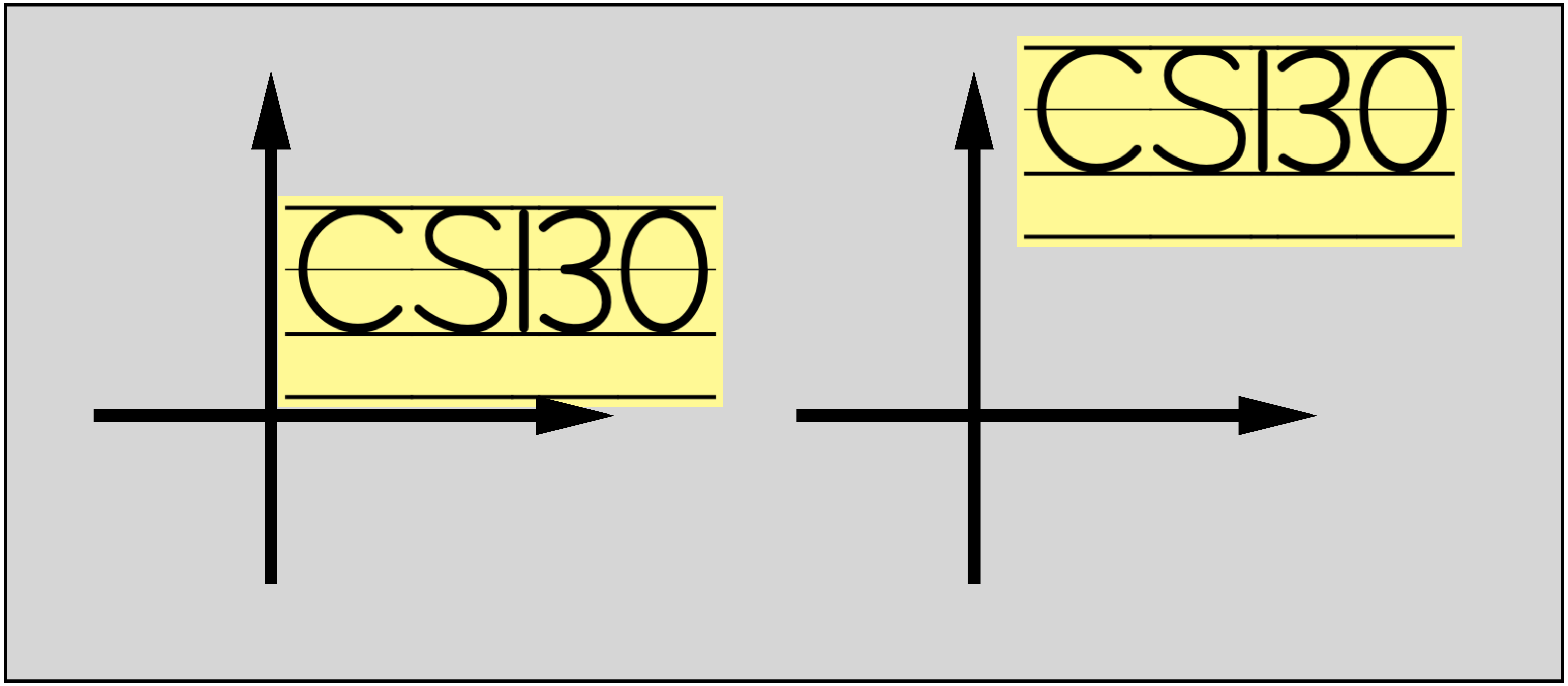
Shear

$$\begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} = \begin{pmatrix} x + ay \\ y \end{pmatrix}$$

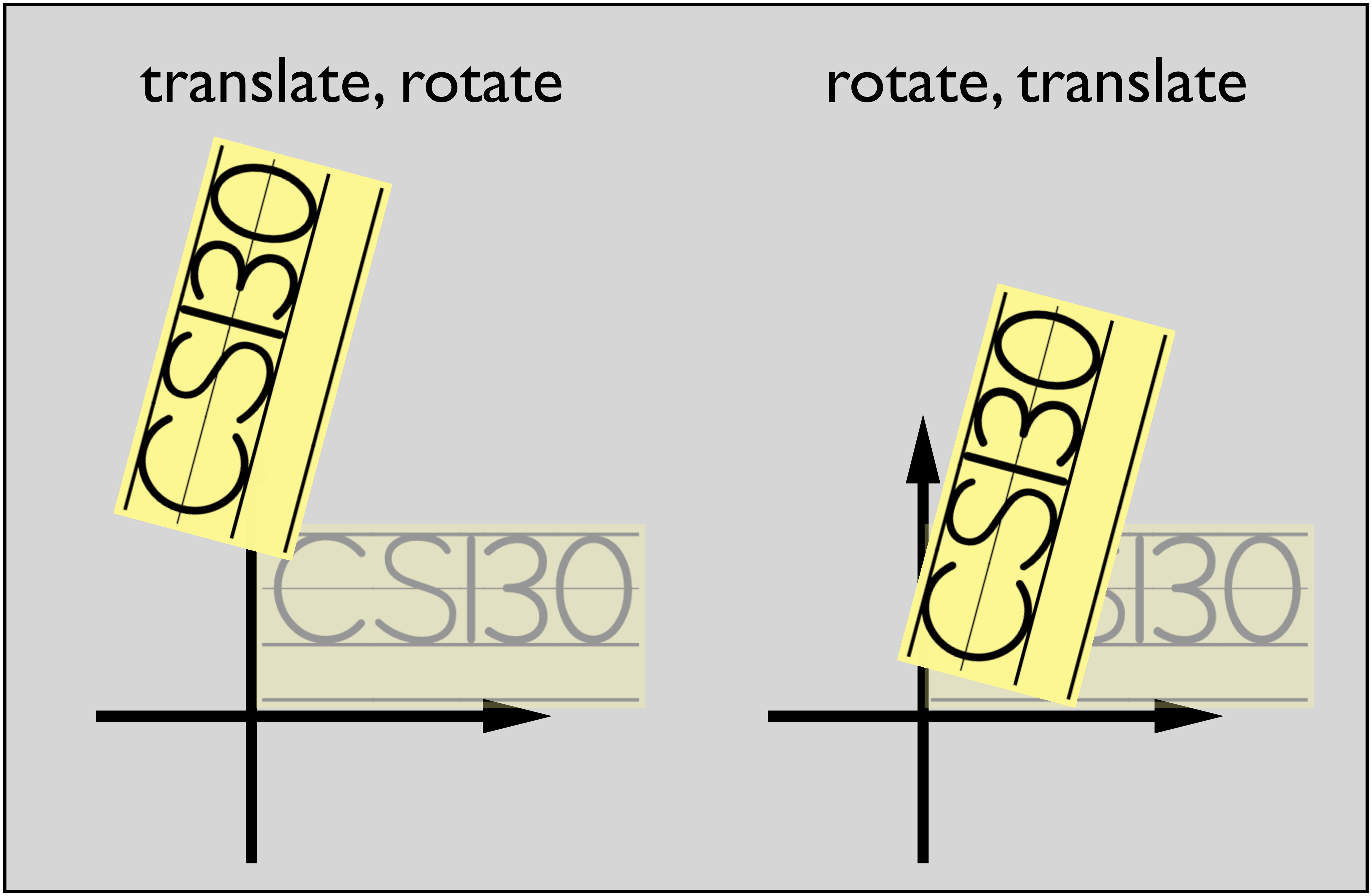


Translation

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



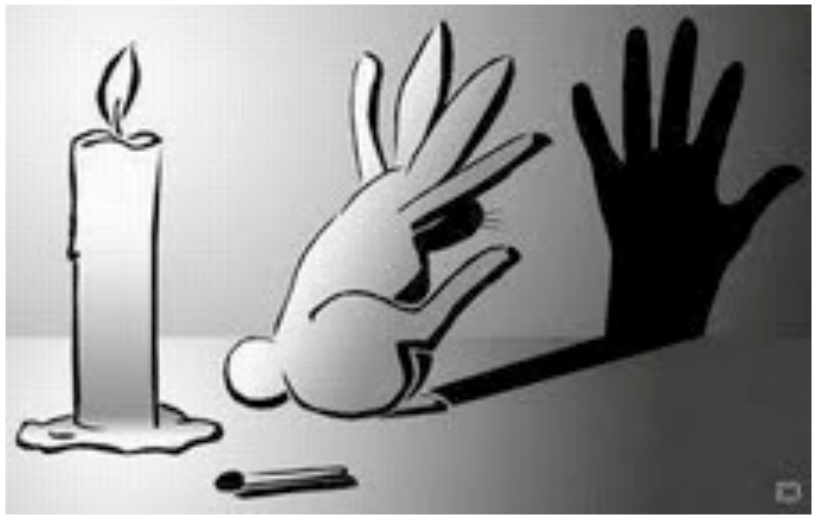
Noncommutativity



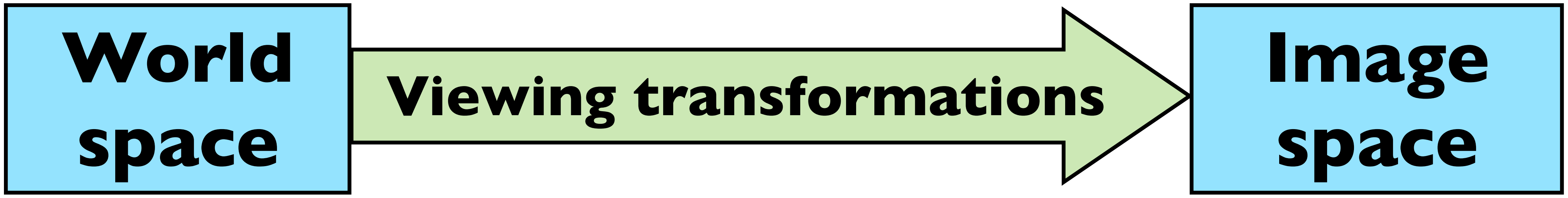
3D Transformations

<whiteboard>

Viewing Transformations



Viewing transformations

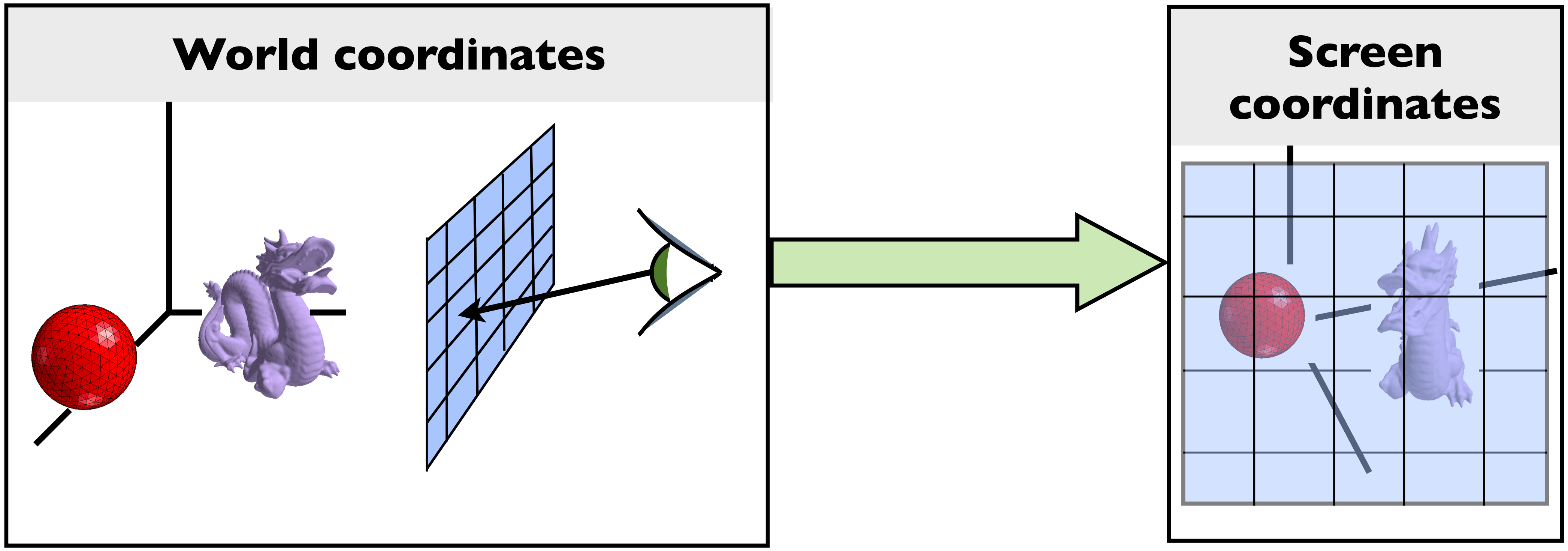


World space

Viewing transformations

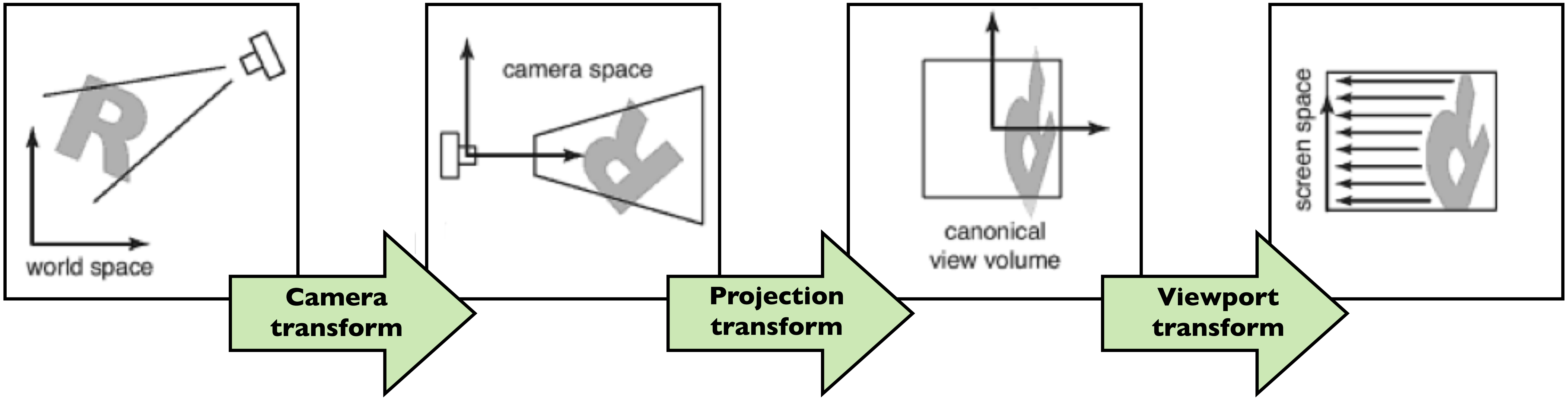
Image space

- Move objects from their 3D locations to their positions in a 2D view



The viewing transformation also projects any point along the pixel's view ray back to the pixel's position in **image space**

Decomposition of viewing transforms



Camera transform

Projection transform

Viewport transform

- •rigid body transformation
- •place camera at origin

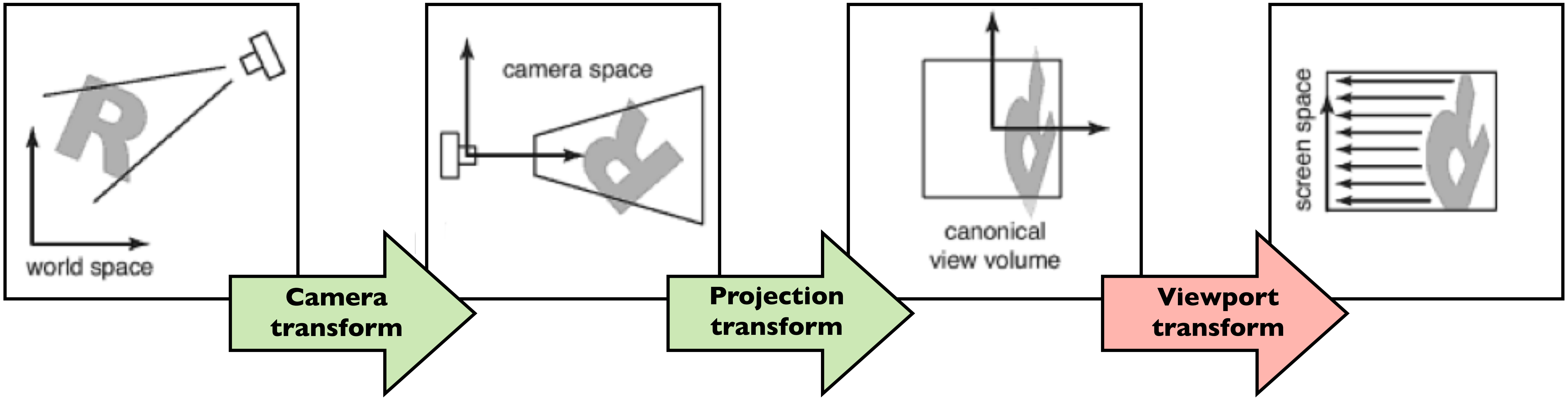
- •x, y, z in [-1,1]
- •depends on type of projection

- •map to pixel coordinates

Viewing transforms depend on: camera position and orientation, type of projection, field of view, image resolution

there are several names for these spaces: “camera space” = “eye space”, “canonical view volume” = “clip space”= “normalized device coordinates”, “screen space=pixel coordinates” and for the transforms: “camera transformation” = “viewing transformation”

Viewport transform



Camera transform

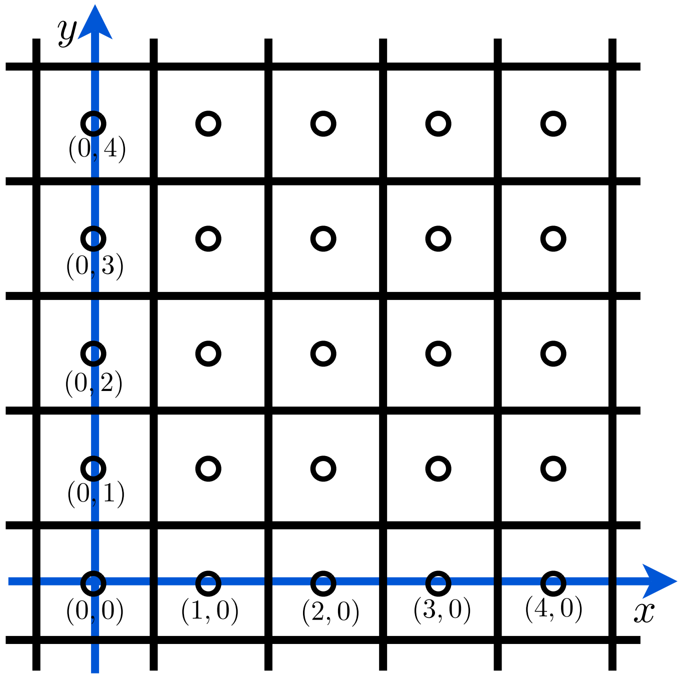
Projection transform

Viewport transform

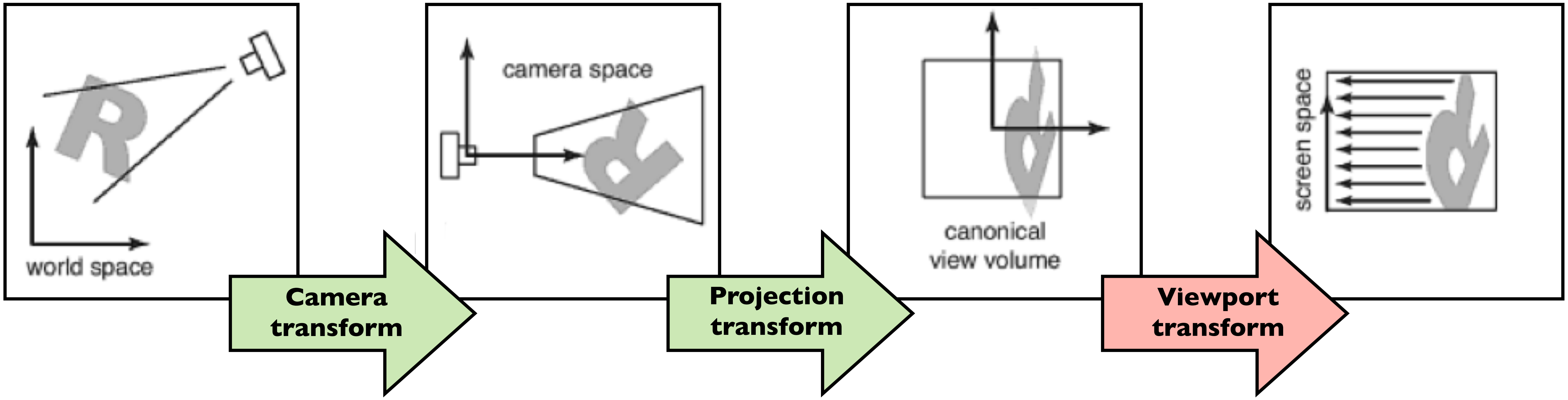
$$(x, y, z) \rightarrow (x', y', z')$$

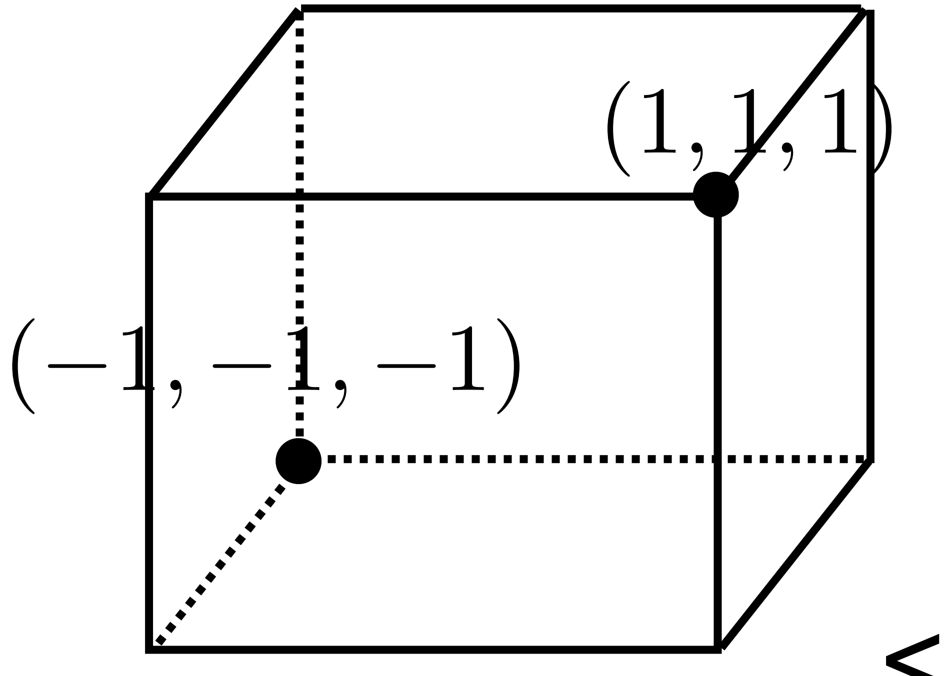
$$(x, y, z) \in [-1, 1]^3$$

$$x' \in [-.5, n_x - .5]$$
$$y' \in [-.5, n_y - .5]$$

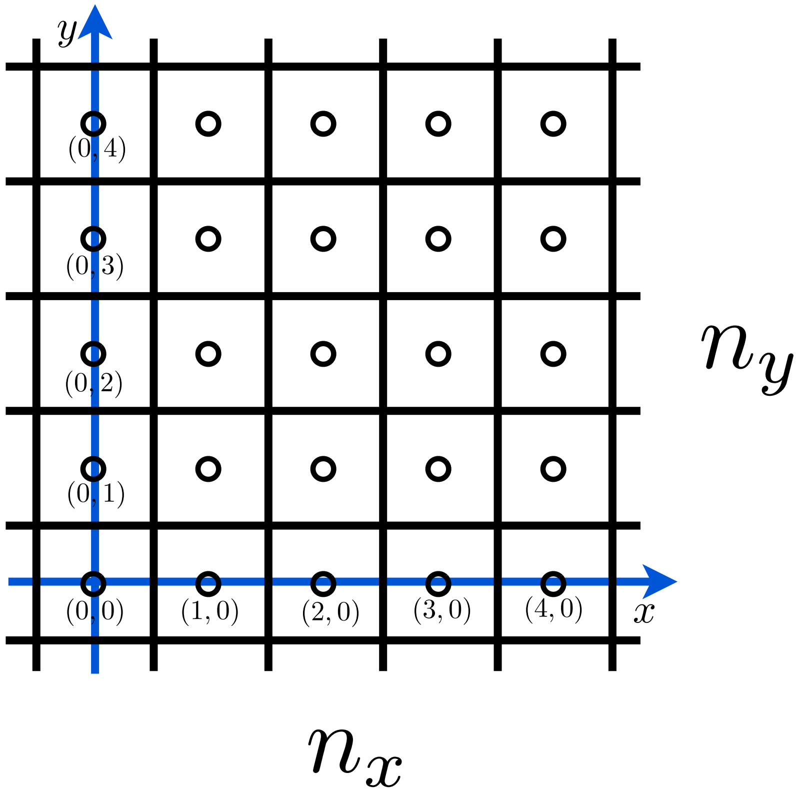


Viewport transform





M_{vp}



<whiteboard>