CSE4203: Computer Graphics

Lecture – 4 (part - B)

Transformation Matrices

Outline

– 3D Linear Transformation
– 3D Scaling
– 3D Rotation
– Translation
– Affine Transformation

3D Transformation (1/1)

- The linear 3D transforms are an extension of the 2D transforms.
  - For 2D:

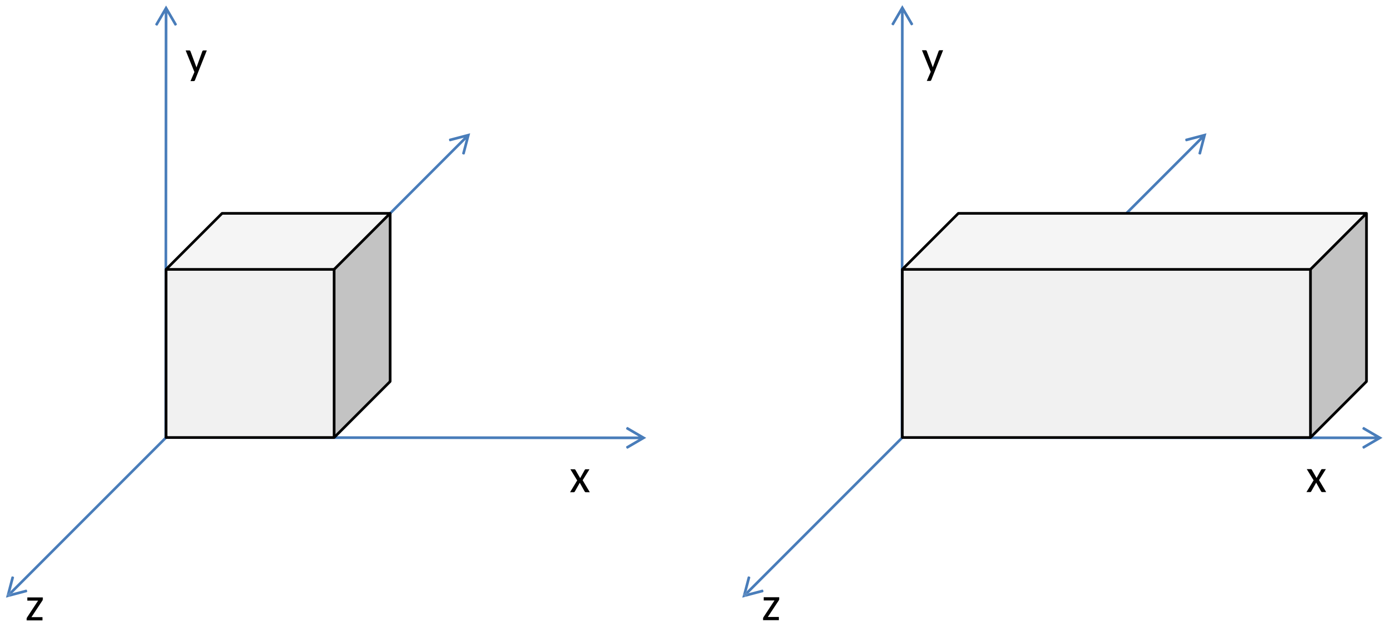
$$\begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} a_{11}x + a_{12}y \\ a_{21}x + a_{22}y \end{bmatrix}$$

  - For 3D:

Credit: Fundamentals of Computer Graphics 3rd Edition by Peter Shirley, Steve Marschner | http://www.cs.cornell.edu/courses/cs4620/2019fa/

3D Scaling (1/1)

$$\text{scale}(s_x, s_y, s_z) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & s_z \end{bmatrix}$$



3D Rotation (1/5)

- Rotation around axis
  - Counter-clockwise, w.r.t rotation axis.

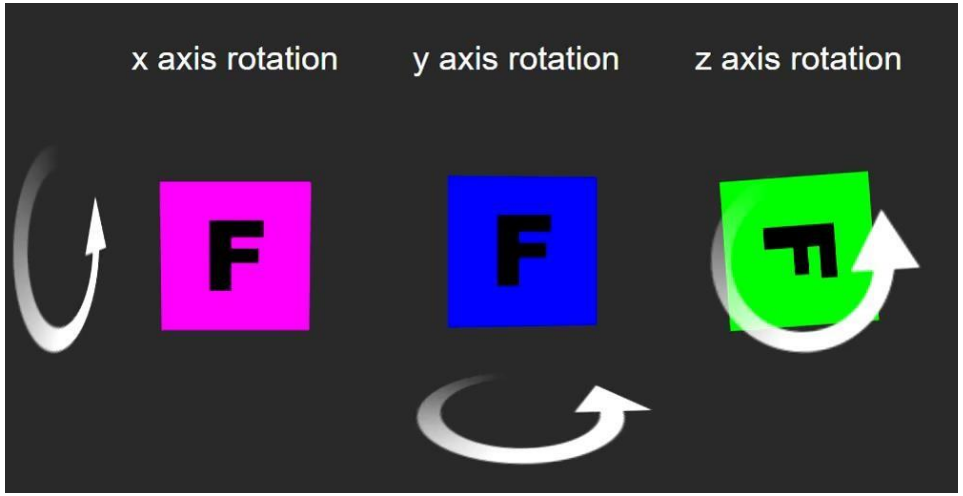
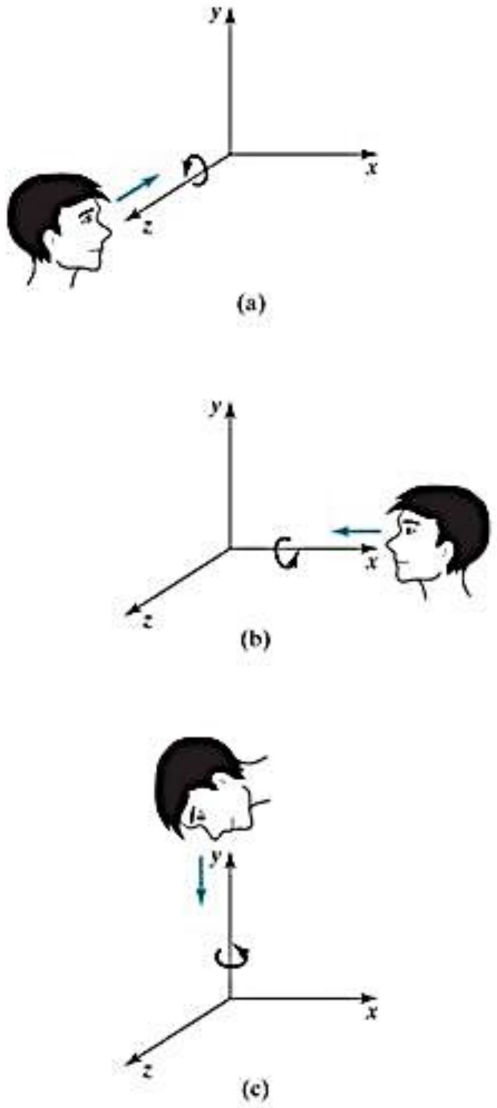
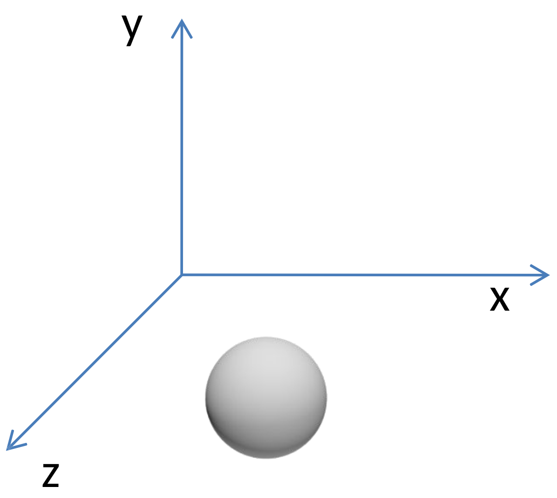


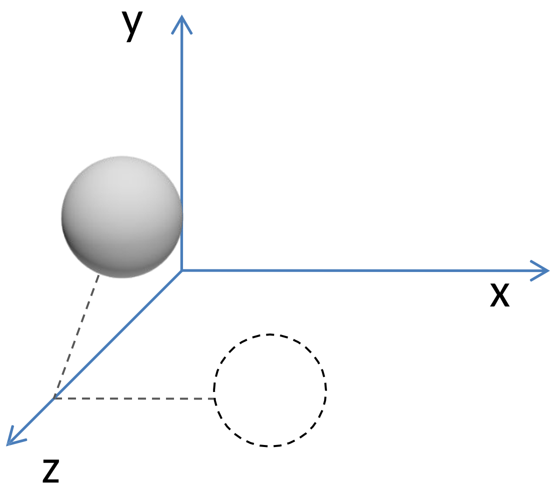
Image Source: https://slideplayer.com/slide/4889962/



3D Rotation (2/5)

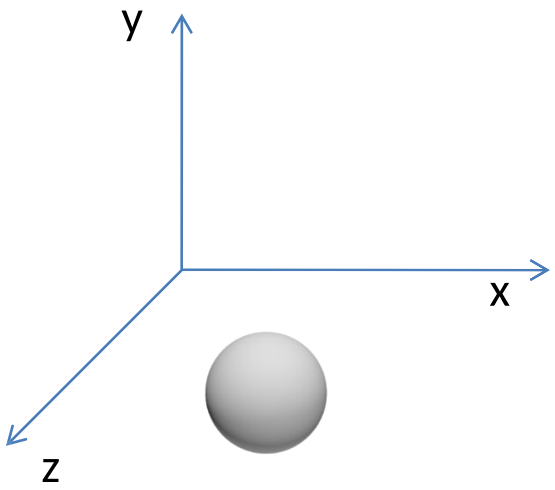
$$\text{rotate-z}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

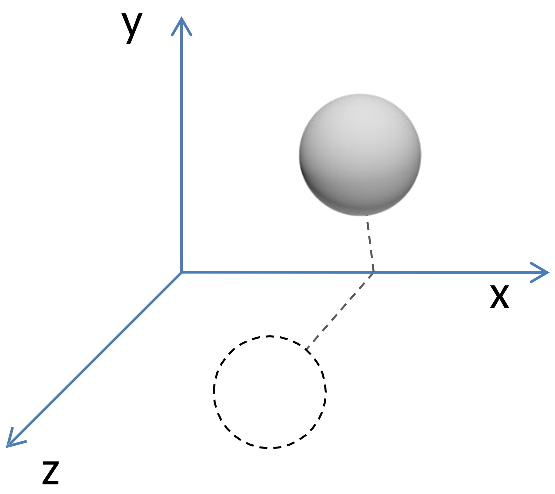




3D Rotation (3/5)

$$\text{rotate-x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$





3D Rotation (4/5)

$$\text{rotate-z}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{rotate-x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

$$\text{rotate-y}(\phi) = \begin{bmatrix} \cos\phi & 0 & \sin\phi \\ 0 & 1 & 0 \\ -\sin\phi & 0 & \cos\phi \end{bmatrix}$$

3D Rotation (5/5)

$$\text{rotate-z}(\phi) = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\text{rotate-x}(\phi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\phi & -\sin\phi \\ 0 & \sin\phi & \cos\phi \end{bmatrix}$$

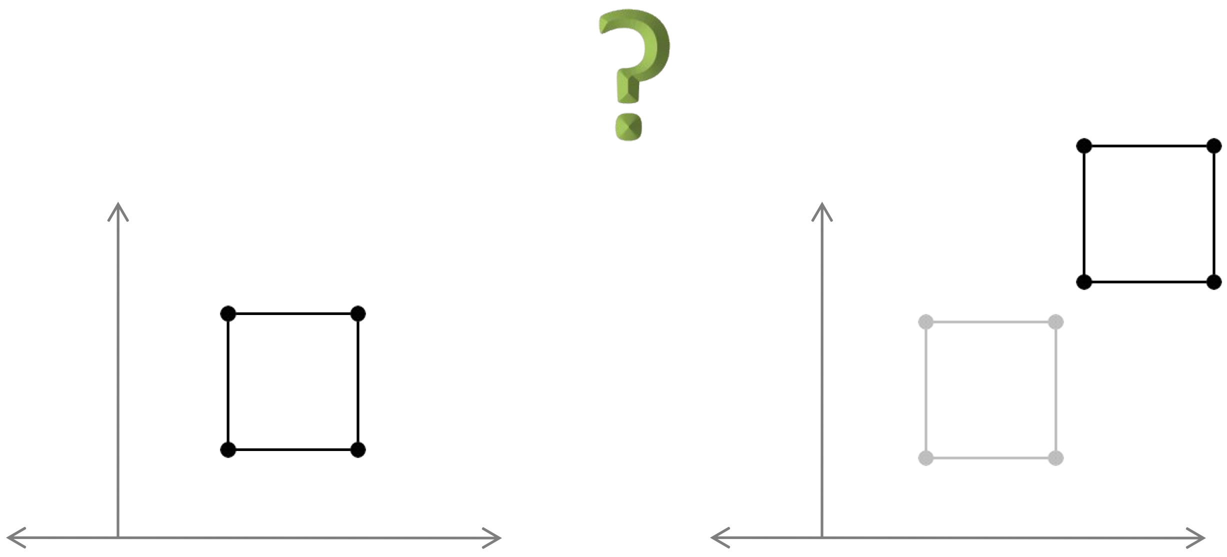
$$\text{rotate-y}(\phi) = \begin{bmatrix} \cos\phi & 0 & \boxed{\sin\phi} \\ 0 & 1 & 0 \\ \boxed{-\sin\phi} & 0 & \cos\phi \end{bmatrix}$$

Q: Why is it different?*

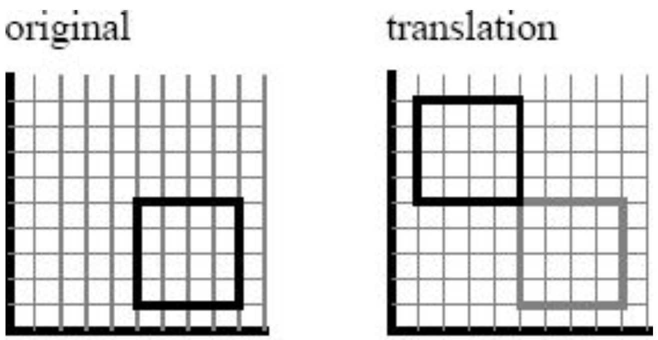
* https://robotics.stackexchange.com/questions/10702/rotation-matrix-sign-convention-confusion

Translation in 2D (1/8)

- Move or Translate to another position.



Translation in 2D (3/8)



$$\underbrace{\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} t_x \\ t_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}}_{}$$

v’ = t + v

$x' = x + t_x$

$y' = y + t_y$

v’ = v + t

Source: https://www.pling.org.uk/cs/cgv.html

Translation in 2D (4/8)

- But, for others cases, i.e. – scaling, rotation, we changed vectors **v** using a **matrix M**.
  - In 2D, these transforms have the form: -

$\begin{aligned} x' &= m_{11}x + m_{12}y, \\ y' &= m_{21}x + m_{22}y. \end{aligned}$	**v’ = M v**

Translation in 2D (5/8)

- We **cannot** use such transforms to **translate**, only to scale and rotate them.

$$\cancel{\begin{aligned} x' &= m_{11}x + m_{12}y, \\ y' &= m_{21}x + m_{22}y. \end{aligned}}$$

$$\cancel{\mathbf{v'} = \mathbf{M}\,\mathbf{v}}$$

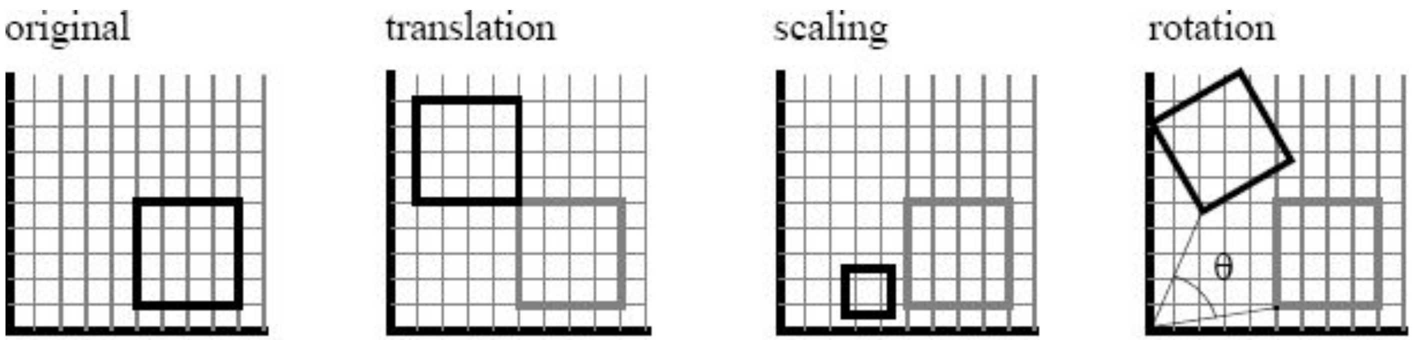
Translation in 2D (6/8)

- There is just no way to do that by **multiplying** (x, y) by a 2 × 2 matrix.
  - **adding** translation to our system of linear transformations:

$\begin{aligned} x' &= m_{11}x + m_{12}y, \\ y' &= m_{21}x + m_{22}y. \end{aligned}$	**v' = M v**
$\begin{aligned} x' &= x + x_t, \\ y' &= y + y_t. \end{aligned}$	**v' = v + t**

Translation in 2D (7/8)

- This is perfectly feasible
  - –



$$\underbrace{\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}t_x\\t_y\end{bmatrix} + \begin{bmatrix}x\\y\end{bmatrix}}_{\mathbf{v' = v + t}}$$

$$\underbrace{\begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}s_x & 0\\0 & s_y\end{bmatrix}\cdot\begin{bmatrix}x\\y\end{bmatrix} \qquad \begin{bmatrix}x'\\y'\end{bmatrix} = \begin{bmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{bmatrix}\cdot\begin{bmatrix}x\\y\end{bmatrix}}_{\mathbf{v' = M\ v}}$$

Source: https://www.pling.org.uk/cs/cgv.html

Translation in 2D (8/8)

- This is perfectly feasible
  - But, the rule for **composing transformations** is not as simple and clean as with linear transformations.

$$T = T_n \,.\, T_{n-1} \,\ldots\, T_1 \,.\, T_0$$

$\begin{aligned} x' &= m_{11}x + m_{12}y, \\ y' &= m_{21}x + m_{22}y. \end{aligned}$	**v' = M v**
$\begin{aligned} x' &= x + x_t, \\ y' &= y + y_t. \end{aligned}$	**v' = v + t**

Homogeneous Coordinates (1/9)

– Instead, we can use a clever trick to get a single matrix multiplication to do both.

$$\begin{bmatrix} x' \\ y' \\ \end{bmatrix} = \begin{bmatrix} & & \\ & \text{2 x 2} & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ \end{bmatrix}$$

Homogeneous Coordinates (2/9)

- Instead, we can use a clever trick to get a single matrix multiplication to do both.

- The idea is simple: represent the point (x, y) by a 3D vector $[x\ y\ 1]^T$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} & & \\ & 3 \times 3 & \\ & & \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates (3/9)

- Instead, we can use a clever trick to get a single matrix multiplication to do both.

- The idea is simple: represent the point (x, y) by a 3D vector **[x y 1]^T**
- Use 3 × 3 matrices of the form.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & x_t \\ m_{21} & m_{22} & y_t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates (4/9)

- This kind of transformation is called an ***affine transformation***.
  - this way of implementing affine transformations by adding an extra dimension is called ***homogeneous coordinates***

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & x_t \\ m_{21} & m_{22} & y_t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates (5/9)

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} m_{11} & m_{12} & x_t \\ m_{21} & m_{22} & y_t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates (6/9)

- Translation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_t \\ 0 & 1 & y_t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates (7/9)

- Scaling:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Homogeneous Coordinates (8/9)

- Rotation:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\varphi & -\sin\varphi & 0 \\ \sin\varphi & \cos\varphi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3D Transformation with Homogeneous Coordinates (1/1)

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

2D/ 3D Transformations (1/3)

	2D	3D
T	$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_t \\ 0 & 1 & y_t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$
S	$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$
R	$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$	$RotX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Theta x) & -\sin(\Theta x) & 0 \\ 0 & \sin(\Theta x) & \cos(\Theta x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $RotY = \begin{bmatrix} \cos(\Theta y) & 0 & \sin(\Theta y) & 0 \\ 0 & 1 & 0 & 0 \\ -\sin(\Theta y) & 0 & \cos(\Theta y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ $RotZ = \begin{bmatrix} \cos(\Theta z) & -\sin(\Theta z) & 0 & 0 \\ \sin(\Theta z) & \cos(\Theta z) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

Inverse Transformations (1/2)

<table>
<tr><th>T</th><th>T^{-1}</th></tr>
<tr><td>$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & p \\ 0 & 1 & 0 & q \\ 0 & 0 & 1 & r \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$</td><td>$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & -p \\ 0 & 1 & 0 & -q \\ 0 & 0 & 1 & -r \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$</td></tr>
<tr><td>$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} p & 0 & 0 & 0 \\ 0 & q & 0 & 0 \\ 0 & 0 & r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$</td><td>$$\begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = \begin{bmatrix} 1/p & 0 & 0 & 0 \\ 0 & 1/q & 0 & 0 \\ 0 & 0 & 1/r & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix}$$</td></tr>
<tr><td>$$RotX = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos(\Theta x) & -sin(\Theta x) & 0 \\ 0 & sin(\Theta x) & \cos(\Theta x) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ $$RotY = \begin{bmatrix} \cos(\Theta y) & 0 & sin(\Theta y) & 0 \\ 0 & 1 & 0 & 0 \\ -sin(\Theta y) & 0 & \cos(\Theta y) & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$ $$RotZ = \begin{bmatrix} \cos(\Theta z) & -sin(\Theta z) & 0 & 0 \\ sin(\Theta z) & \cos(\Theta z) & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$</td><td>?</td></tr>
</table>

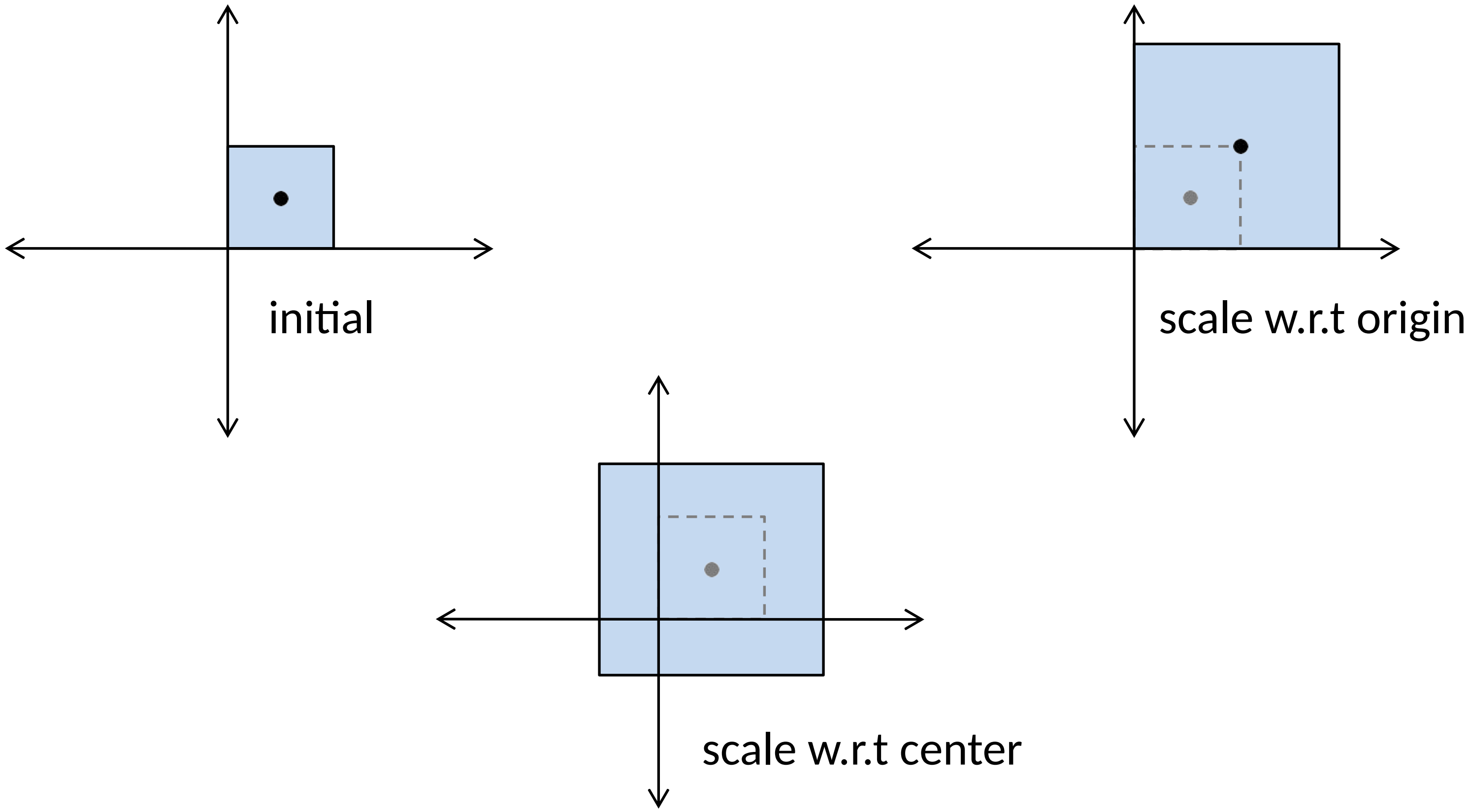
Inverse Transformations (2/2)

	Transformation	Inverse Transformation
T	T (tx, ty, tz)	T^{-1} = T$(-tx, -ty, -tz)$
S	S (sx, sy, sz)	S^{-1} = S $(1/sx, 1/sy, 1/sz)$
R	Rx(d) Ry(d) Rz(d)	R^{-1} = R$(-d)$ = R^T Rx^{-1} = Rx^T Ry^{-1} = Ry^T Rz^{-1} = Rz^T

<u>**Task:**</u> take any transformation matrix (i.e. scaling matrix ***S***) with numerical values, do the matrix inversion and see if it becomes S^{-1}

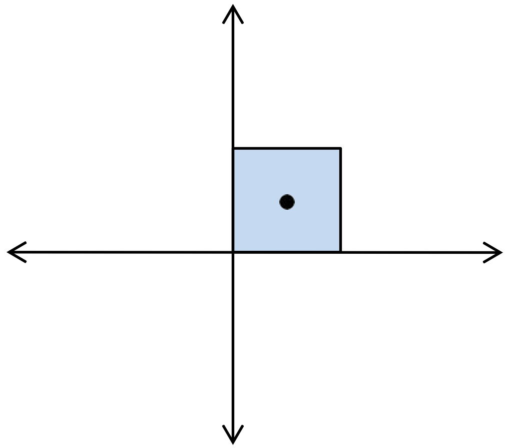
Practice Problem - 1

- Scale w.r.t the center



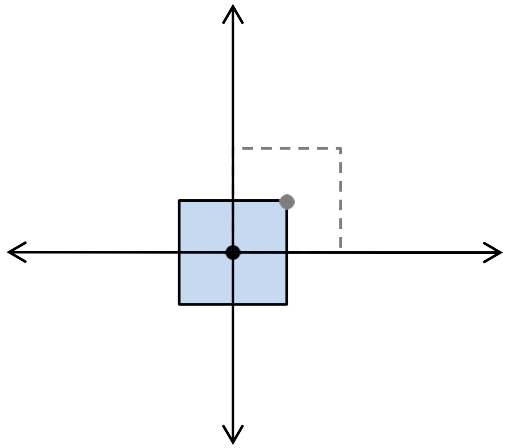
Practice Problem – 1 (Sol.)

- Scale w.r.t the center



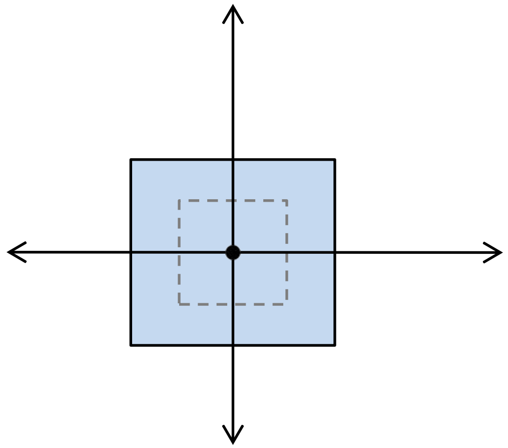
Practice Problem – 1 (Sol.)

- Scale w.r.t the center



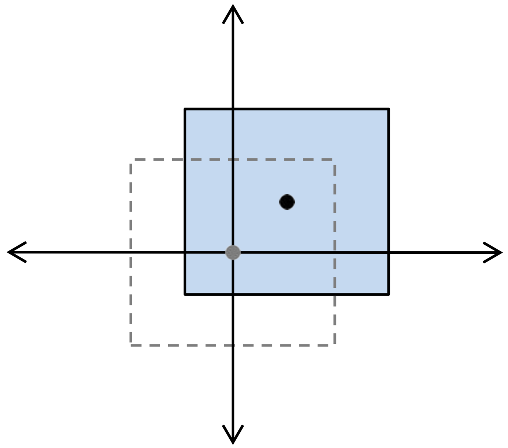
Practice Problem – 1 (Sol.)

- Scale w.r.t the center



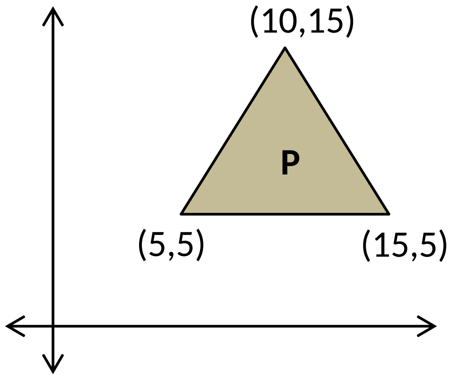
Practice Problem – 1 (Sol.)

- Scale w.r.t the center



Practice Problem - 2

- We need to rotate a pyramid ***P*** about point **(5, 5)** by **90º**. You have to **–**
  - Mention the steps to perform the task.
  - Determine the composite transformation matrix ***M***.
  - Multiply ***M*** with ***P*** and determine the new coordinates ***P'***.
  - Plot ***P*** and ***P'*** on the same axis to show the rotation.



Further Reading

- Fundamentals of Computer Graphics, 4th Edition - Chapter 6 (Exercise 1 – 6, 8 and 9)