Computer Graphics: CO 303

Lecture 4, 5

# Draconifors

## Homogenous coordinates

Zubin Bhuyan,

Department of CSE, Tezpur University

http://www.tezu.ernet.in/~zubin

# Transformation as Matrices

$$\begin{matrix} s_x & 0 \\ 0 & s_y \end{matrix} \quad \begin{matrix} x \\ y \end{matrix} = \begin{matrix} s_x \cdot x \\ s_y \cdot y \end{matrix}$$

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

$$\begin{matrix} t_x \\ t_y \end{matrix} + \begin{matrix} x \\ y \end{matrix} = \begin{matrix} x + t_x \\ y + t_y \end{matrix}$$

# Transformation as Matrices

$$\begin{matrix} s_x & 0 \\ 0 & s_y \end{matrix} \quad \begin{matrix} x \\ y \end{matrix} = \begin{matrix} s_x \cdot x \\ s_y \cdot y \end{matrix}$$

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

$$\begin{matrix} t_x \\ t_y \end{matrix} + \begin{matrix} x \\ y \end{matrix} = \begin{matrix} x + t_x \\ y + t_y \end{matrix}$$

*Only translation cannot be expressed as matrix-vector multiplications.*

# Homogenous Coordinates

- In order to represent translation as a matrix multiplication operation we use 3 x 3 matrices and pad our points to become 3 x 1 matrices.

$$Trans(d_x, d_y) = \begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \quad Scale(s_x, s_y) = \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$Rot(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad Shear(a_x) = \begin{bmatrix} 1 & a_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Connection to 3D?
- (x, y, 1) represents a 3D point on the plane W = 1.
*A homogeneous point is a* ***line in 3D, through the origin***

# Composite Transformations

- Suppose we wish to perform multiple transformations on a point:

$$P_2 = T_{3,1} P_1$$

$$P_3 = S_{2,2} P_2$$

$$P_4 = R_{30} P_3$$

- But we want something like this:

$$M = R_{30} S_{2,2} T_{3,1}$$

$$P_4 = M P_1$$

The first transformation you want to perform will be at the far right, just before the point.
*[Matrix multiplication is associative, not commutative!]*

# Composite Transformations

- Suppose we wish to perform multiple transformations on a point:

$$P_2 = T_{3,1}P_1$$

$$P_3 = S_{2,2}P_2$$

$$P_4 = R_{30}P_3$$

- But we want something like this:
  - Concatenate basic transforms sequentially.

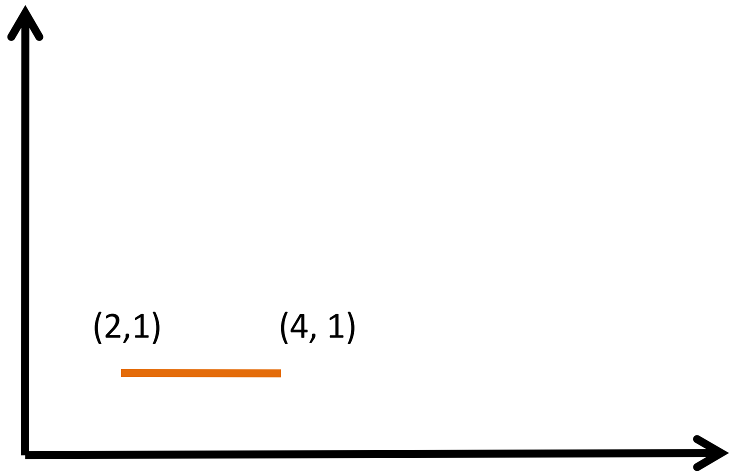
$$M = R_{30}S_{2,2}T_{3,1}$$

$$P_4 = MP_1$$

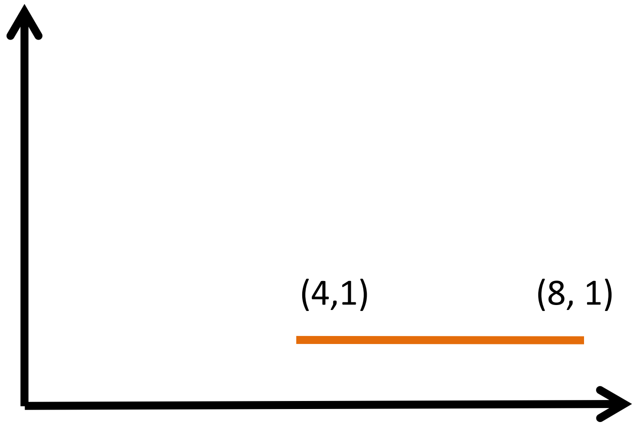
The first transformation you want to perform will be at the far right, just before the point.
*[Matrix multiplication is associative, not commutative!]*

# Example: Fixed Point Scaling

- Whenever scale transformation is performed, it also moves the object being scaled.
  - Example: Scale a line between (2, 1) (4,1) to twice its length.



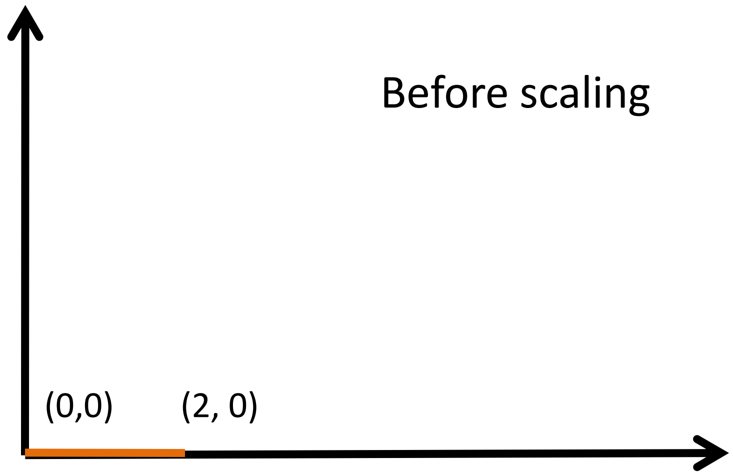

Original

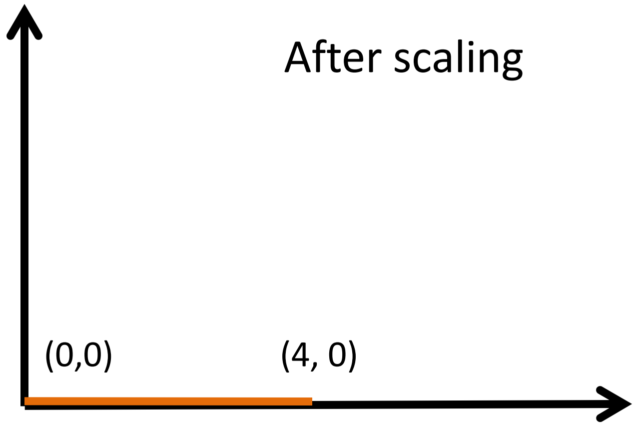



After scaling

# Example: Fixed Point Scaling

- If we scale a line between (0,0) & (2,0) to twice its length, the left-hand endpoint does not move.

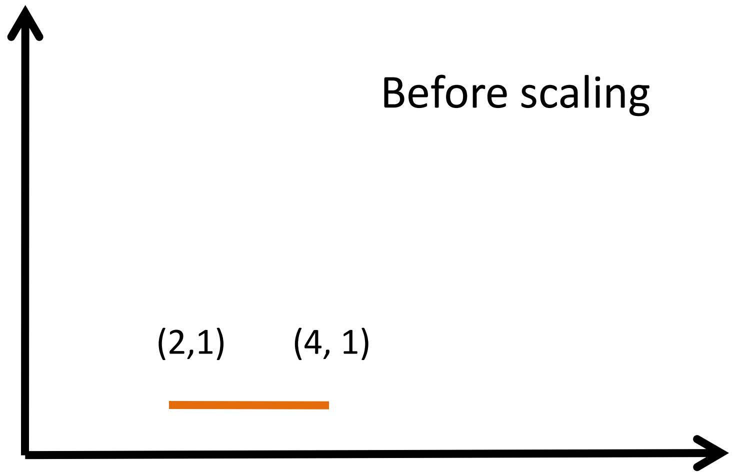


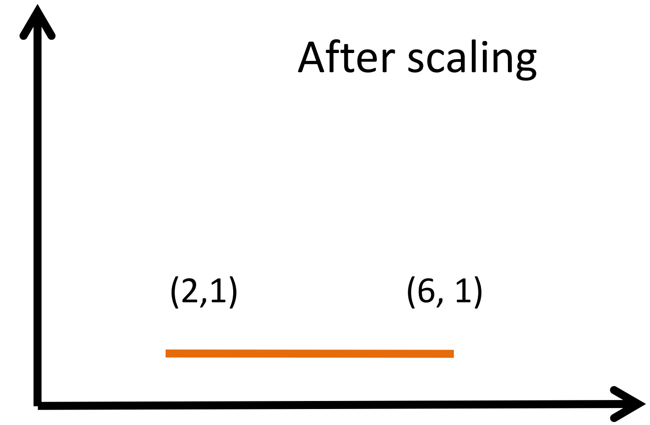




(0,0) is known as a fixed point for the basic scaling transformation.
We can use composite transformations to create a scale transformation with different fixed points.

# Example: Fixed Point Scaling

- Scale by 2 with fixed point = (2,1)
- Translate the point (2,1) to the origin
- Scale by 2
- Translate origin to point (2,1)

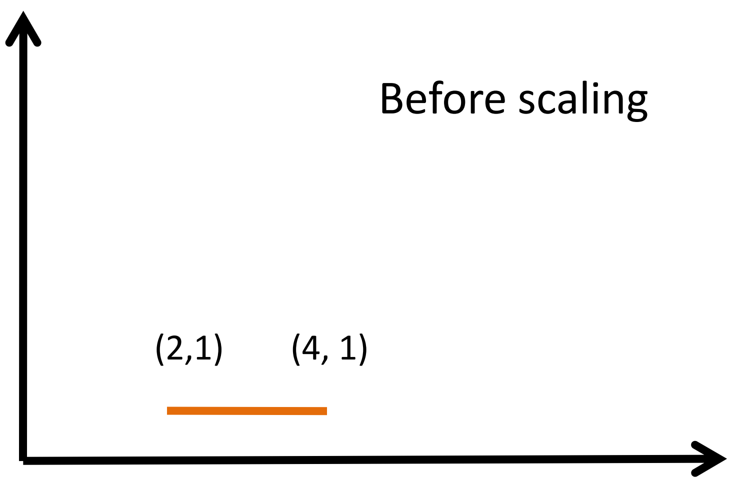


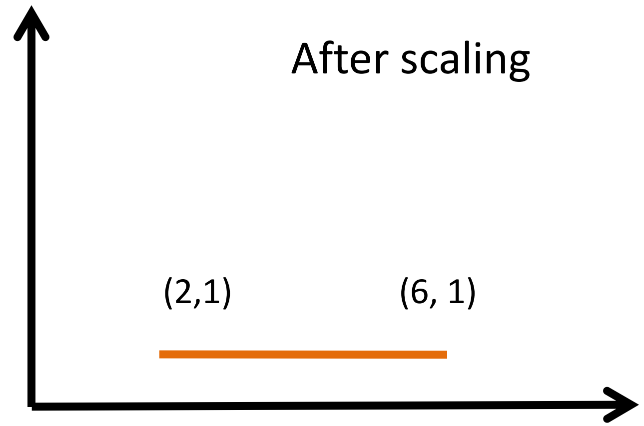



# Example: Fixed Point Scaling

$$\underbrace{\begin{bmatrix}1 & 0 & 2\\0 & 1 & 1\\0 & 0 & 1\end{bmatrix}}_{T_{2,1}}\underbrace{\begin{bmatrix}2 & 0 & 0\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}}_{S_{2,1}}\underbrace{\begin{bmatrix}1 & 0 & -2\\0 & 1 & -1\\0 & 0 & 1\end{bmatrix}}_{T_{-2,-1}}=\underbrace{\begin{bmatrix}2 & 0 & -2\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}}_{C}$$

$$\underbrace{\begin{bmatrix}2 & 0 & -2\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}}_{C}\begin{bmatrix}2\\1\\1\end{bmatrix}=\begin{bmatrix}2\\1\\1\end{bmatrix} \qquad \underbrace{\begin{bmatrix}2 & 0 & -2\\0 & 1 & 0\\0 & 0 & 1\end{bmatrix}}_{C}\begin{bmatrix}4\\1\\1\end{bmatrix}=\begin{bmatrix}6\\1\\1\end{bmatrix}$$



# Compound Translation

- What happens when a point goes through

  $T(d_{x1}, d_{y1})$ and then $T(d_{x2}, d_{y2})$?

- Combined translation: $T(d_{x1}+d_{x2}, d_{y1}+d_{y2})$

$$\begin{bmatrix} 1 & 0 & d_{x_2} \\ 0 & 1 & d_{y_2} \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & d_{x_1} \\ 0 & 1 & d_{y_1} \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & d_{x_1} + d_{x_2} \\ 0 & 1 & d_{y_1} + d_{y_2} \\ 0 & 0 & 1 \end{bmatrix}$$

Concatenation of transformations: **matrix multiplication**

# Compound Rotation

- What happens when a point goes through

  R(θ) and then R(ϕ)?

- Combined rotation: R(θ + ϕ)

$$\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\phi & -\sin\phi & 0 \\ \sin\phi & \cos\phi & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta+\phi) & -\sin(\theta+\phi) & 0 \\ \sin(\theta+\phi) & \cos(\theta+\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

- Scaling or shearing (in one direction) is similar
- These concatenations are **commutative**

# Commutativity

- Transformations that commute
  - Translate-Translate
  - Rotate-Rotate (around the same axis)
  - Scale-Scale
  - Uniform scale-Rotate
  - Shear in X (or Y) - Shear in X (or Y)

- *The first transformation you want to perform will be at the far right, just before the point.*

# Inverse of Transformations

- Inverse of $T(d_x, d_y) = T(-d_x, -d_y)$

$$\begin{bmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$T(-d_x, -d_y) = T(d_x, d_y)^{-1}$$

- $R(\theta)^{-1} = R(-\theta)$
- $S(s_x, s_y)^{-1} = S(1/s_x, 1/s_y)$
- $Sh(a_x)^{-1} = Sh(-a_x)$

# Example: Rotation about arbitrary axis

THANKS!