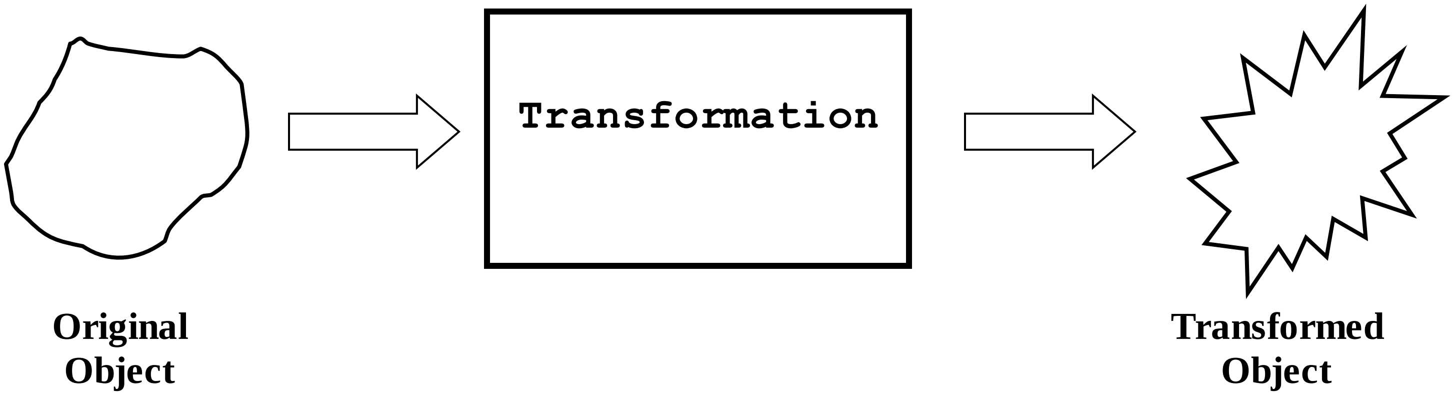
CSc 133 Lecture Notes

13 - Transformations

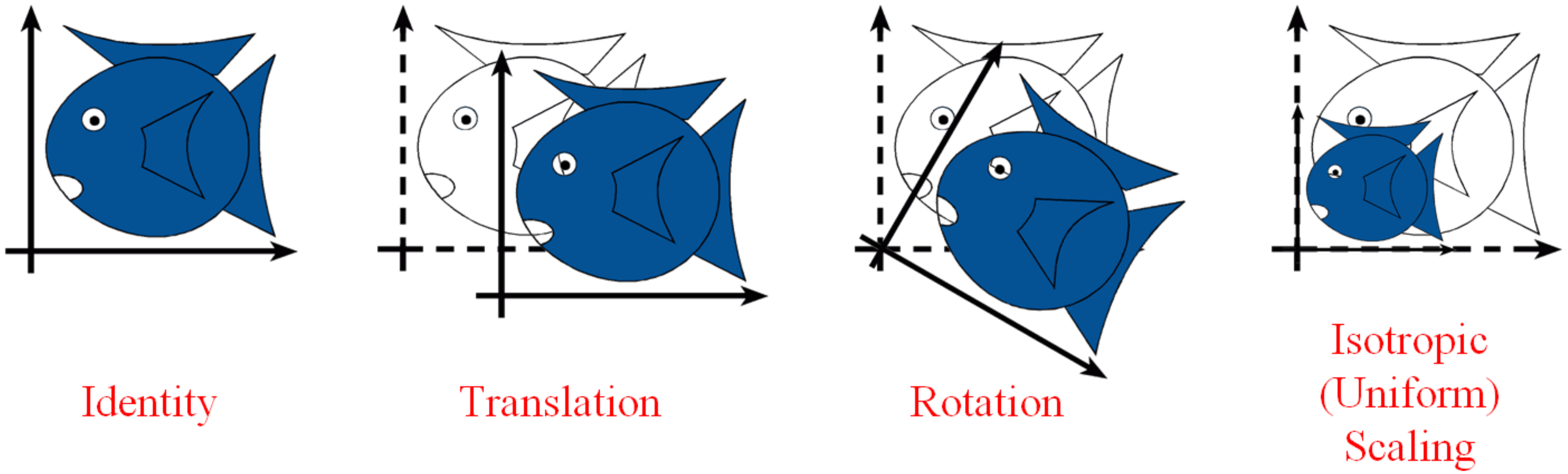
Computer Science Department
California State University, Sacramento

The “Transformation” Concept



- “Original object” could be anything
  - We will focus on geometric objects
- “Transformed object” is usually (*but not necessarily*) of same type

Simple Transformations



- Can be combined
- Are these operations invertible?

  Yes, except scale = 0

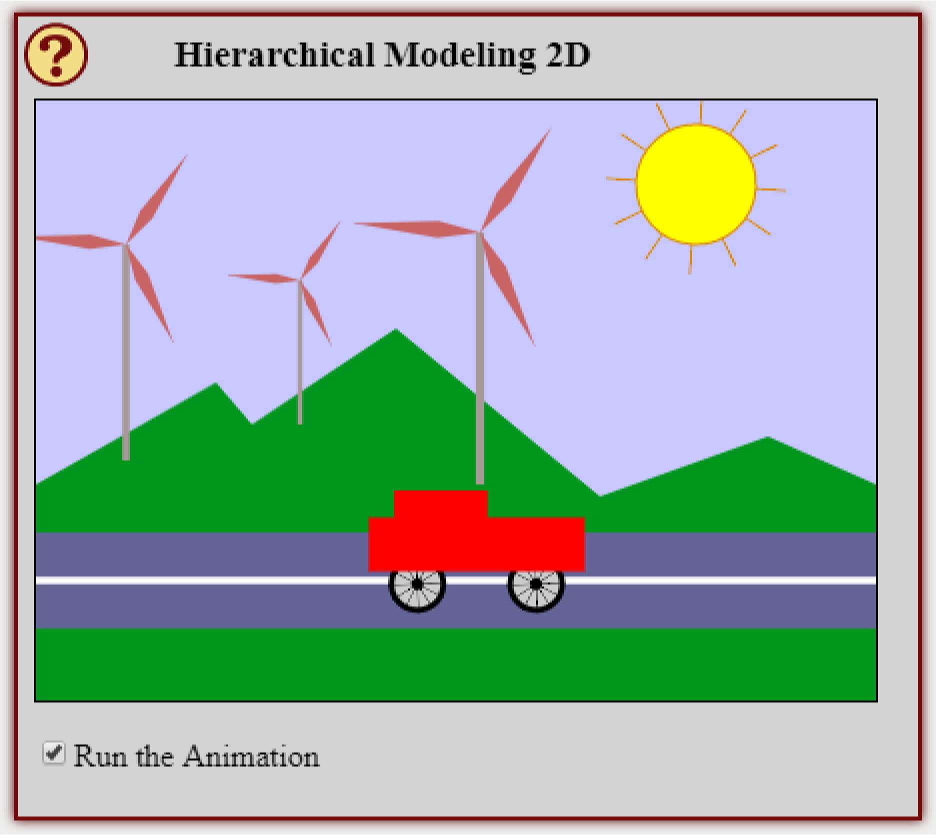
Credits: MIT EECS 6.837, Durand and Cutler

Transformations are used:

- Position objects in a scene (modelling)
- Change the shape of objects
- Create multiple copies of objects
- Projection for virtual cameras
- Animations

Credits: MIT EECS 6.837, Durand and Cutler

Example: an animated scene



You can probably guess how hierarchical modeling is used to draw the three windmills in this example. There is a *drawWindmill* method that draws a windmill in its own coordinate system. Each of the windmills in the scene is then produced by applying a different modeling transform to the standard windmill. Furthermore, the windmill is itself a complex object that is constructed from several sub-objects using various modeling transformations.

Credits: http://math.hws.edu/graphicsbook/c2/s4.html

Overview

- **Part 1:** translate, rotate scale and reflect objects using matrices (mathematics background - Today)
- **https://www.youtube.com/watch?v=DD70ZIDjL7g**

Overview

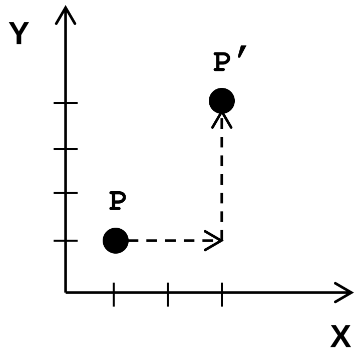
- **Part 2:**
- **Affine Transformations: Translation, Rotation, Scaling**
- **Transforming Points & Lines**
- **Matrix Representation of Transforms**
- **Homogeneous Coordinates**
- **Concatenation of Transformations**

“Affine” Transformations

- Properties:
    - “Map” (transform) finite points into finite points
    - Map parallel lines into parallel lines
- Common examples used in graphics:
    - Translation
    - Rotation
    - Scaling

Transformations on Points

- Translation



P = (x, y)

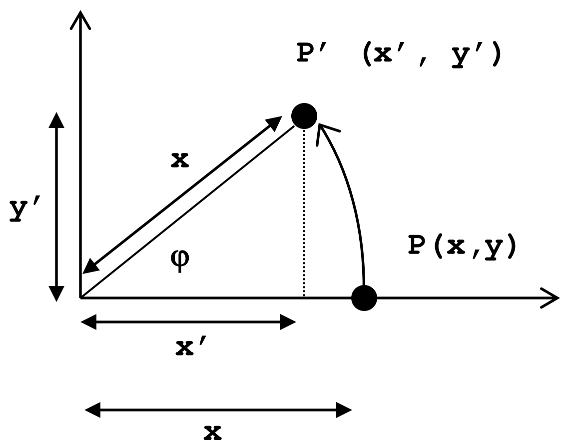
T = (+2, +3)

P′ = (x+2, y+3)

$P \rightarrow \boxed{T} \rightarrow P'$ or $P' \leftarrow \boxed{T} \leftarrow P$

Transformations on Points (cont.)

- Rotation about the origin (point on X axis)



$\cos(\varphi) = x' / x$; hence

$$x' = x \cos(\varphi)$$

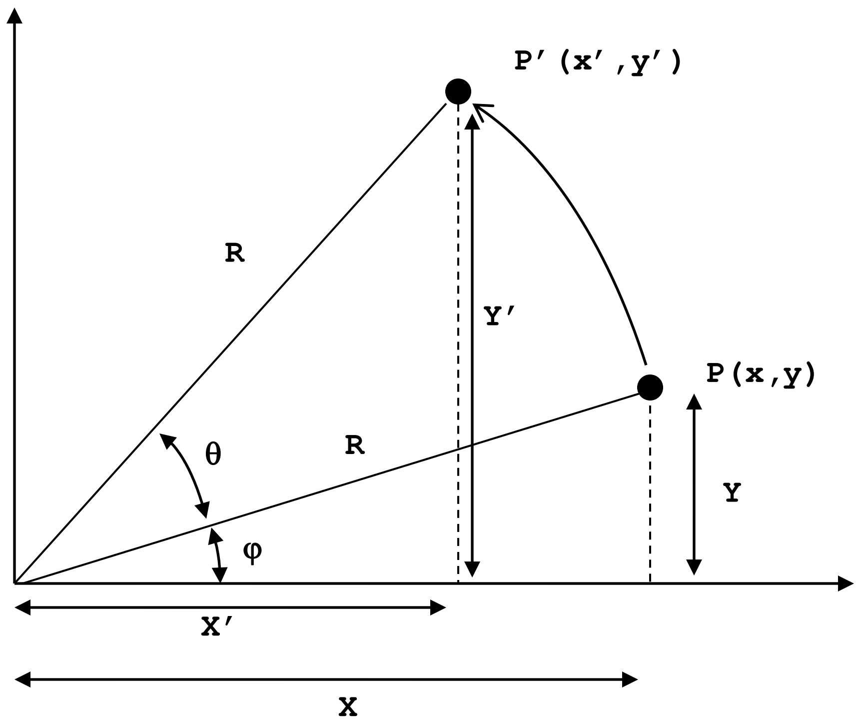
$\sin(\varphi) = y' / x$; hence

$$y' = x \sin(\varphi)$$

$P \rightarrow \boxed{R} \rightarrow P'$ or $P' \leftarrow \boxed{R} \leftarrow P$

Transformations on Points (cont.)

- Rotation about the origin (arbitrary point)



Sum-Difference Formulas

$$\sin(u \pm v) = \sin u \cos v \pm \cos u \sin v$$
$$\cos(u \pm v) = \cos u \cos v \mp \sin u \sin v$$

$\cos(\varphi) = X / R$ and $\sin(\varphi) = Y / R$;

$X = R\cos(\varphi)$ and $Y = R\sin(\varphi)$

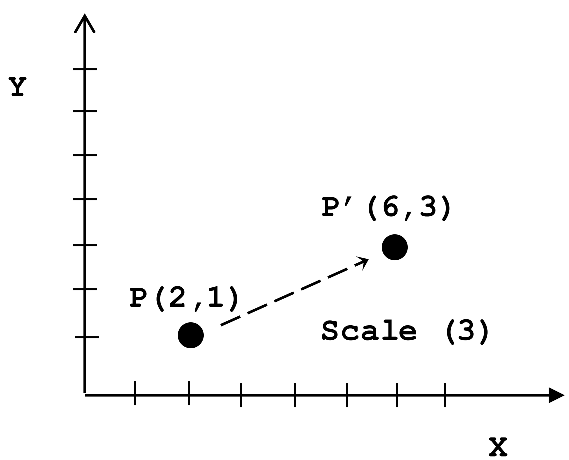
$$
\begin{aligned}
X' &= R\cos(\varphi + \theta) \\
&= R\,(\cos(\varphi)\cos(\theta) - \sin(\varphi)\sin(\theta)) \\
&= \underline{R\cos(\varphi)}\,\cos(\theta) - \underline{R\sin(\varphi)}\,\sin(\theta) \\
&= \underline{X}\cos(\theta) - \underline{Y}\sin(\theta)
\end{aligned}
$$

Similarly,

$$Y' = X\sin(\theta) + Y\cos(\theta)$$

Transformations on Points (cont.)

- Scaling
  - Multiplication by a “scale factor”



$P = (x, y)$

$S = (s_x, s_y)$

$P' = (x*s_x, y*s_y)$

$P \rightarrow \boxed{S} \rightarrow P'$ or $P' \leftarrow \boxed{S} \leftarrow P$

Transformations on Points (cont.)

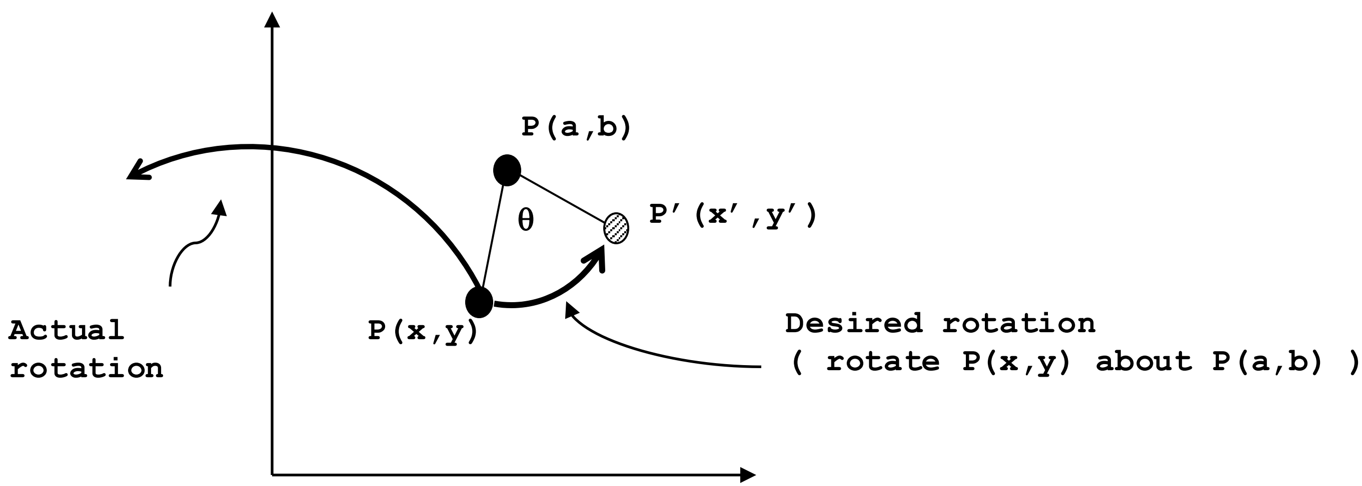
- Scaling is
  - Relative to the origin (like rotation)
  - *Different* from a "move":
    - Translate (3,3) always moves exactly 3 units
    - Scale (3,3) depends on the initial point being scaled:

```
P(1,1)*Scale(3,3) → P'(3,3)    ("move" of 2)
P(4,4)*Scale(3,3) → P'(12,12)  ("move" of 8)
```

- Scaling by a fraction: move "closer to origin"
- Scaling by a negative value:
  "reflection" across axes ("mirroring")
- Scaling where $s_x \neq s_y$: change "aspect ratio"

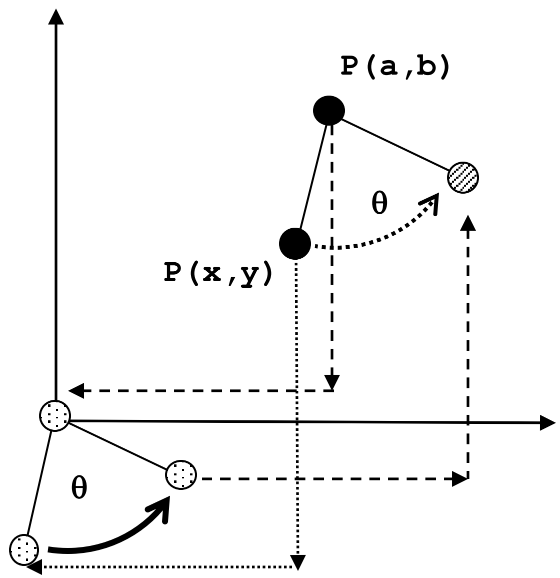
Transformations on Points (cont.)

- Rotating a point about an arbitrary point
  - Problem: rotation formulas are ***relative to the origin***



Transformations on Points (cont.)

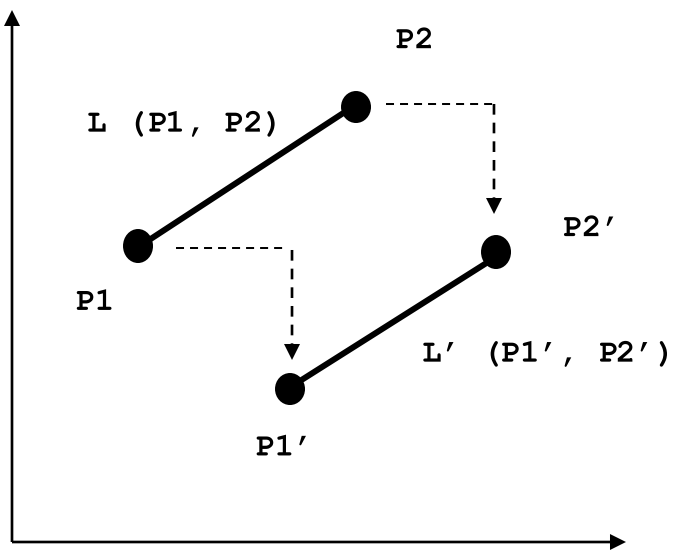
- Solution:
  - Translate to origin
  - Perform rotation
  - Translate "back"



1. **Translate P(x,y) by (-a, -b)**
2. **Rotate (translated) P**
3. **"Undo" the translation**
   (translate result by (+a, +b))

Transformations on Lines

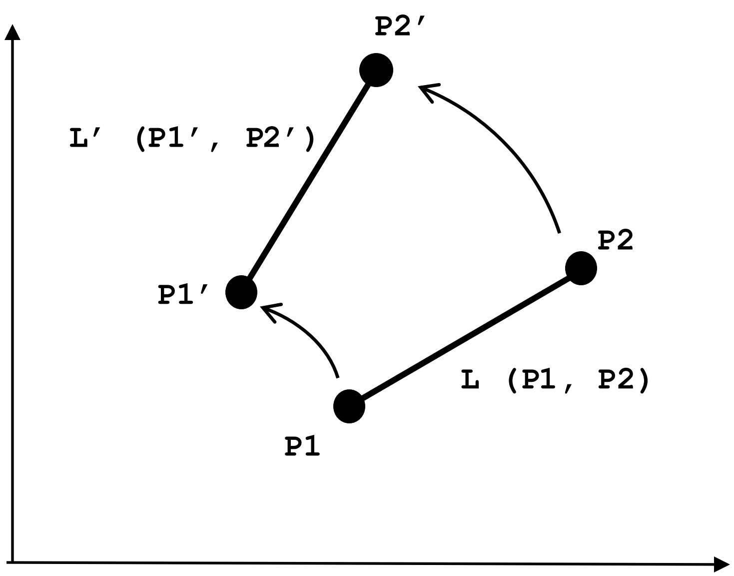
- Translation: translate the endpoints



- **Translate (Line(p1,p2))**
  = Line (Translate(p1), Translate(p2))

Transformations on Lines (cont.)

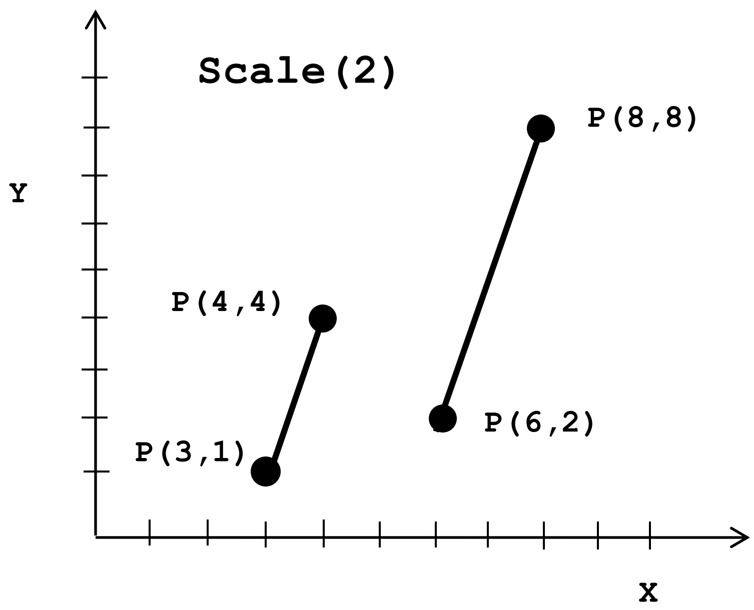
- Rotation *about the origin*: rotate the endpoints



- **Rotate (Line(p1,p2))**
  = Line (Rotate(p1), Rotate(p2))

Transformations on Lines (cont.)

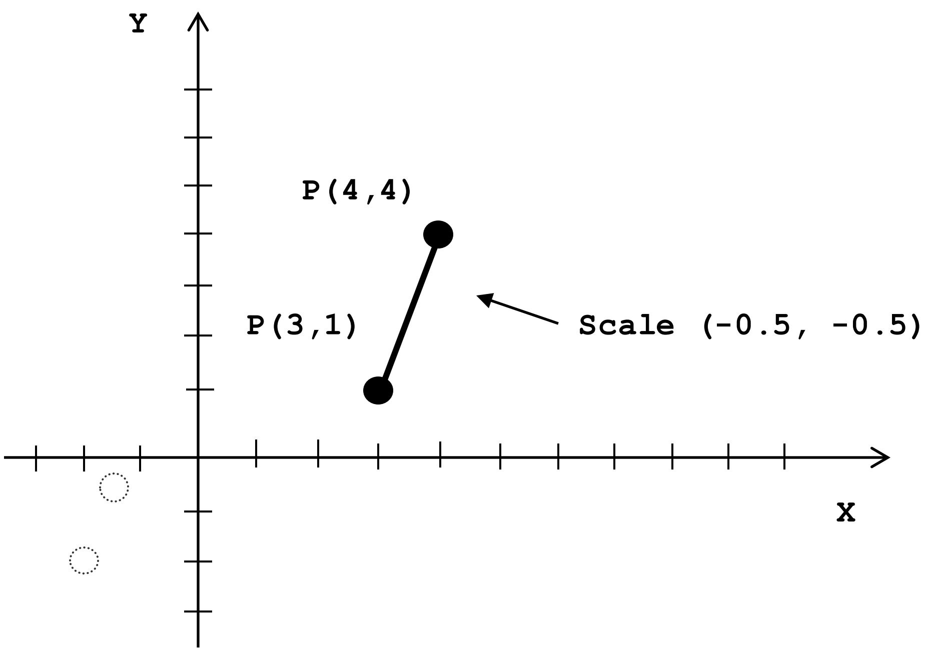
- Scaling: scale the endpoints



- Scale (Line(p1,p2))
  = Line (Scale(p1), Scale(p2))
- Note how scale seems to "move" also

Transformations on Lines (cont.)

- Question: what is the result of
  `Scale(-0.5, -0.5)` applied to this line?



Some general rules for <u>scaling</u>:

- Absolute Value of Scale Factor > 1 → "bigger"
- Absolute Value of Scale Factor < 1 → "smaller"
- Scale Factor < 0 → "flip" ("mirror")

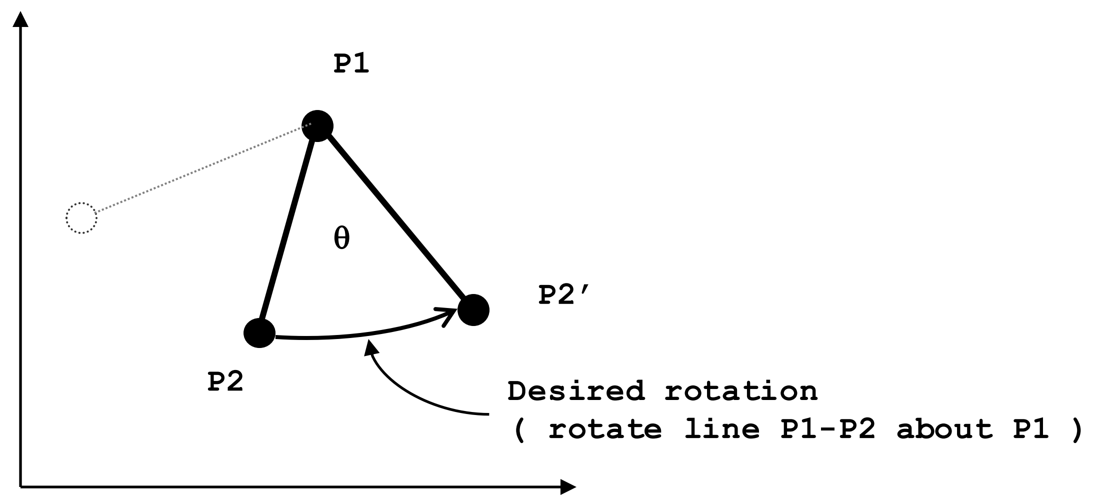
Identity Operations:

- For translation: 0 → No Change
- For rotation: 0 → No Change
- For scaling: 1 → No Change

Transformations on Lines (cont.)

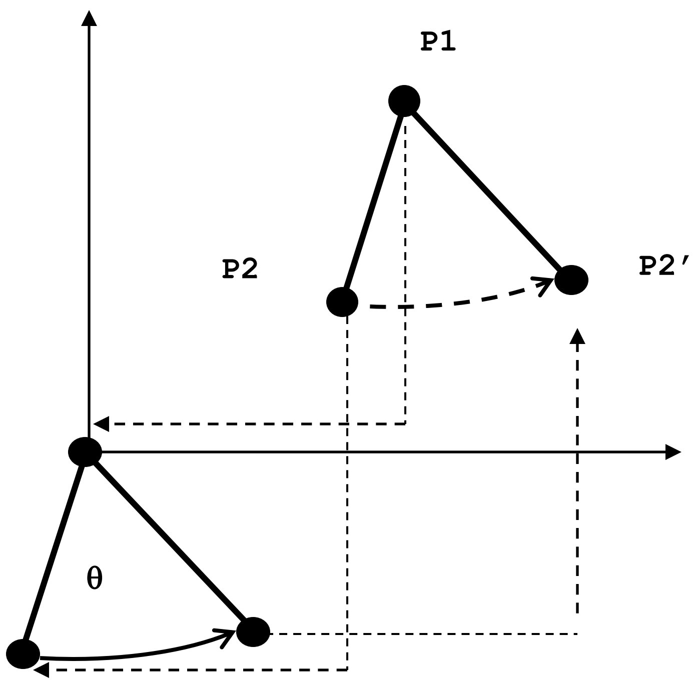
- Rotating a line about an endpoint
  - Intent: P1 doesn't change, while P2 → P2'

    (i.e. *rotate P2 by θ about P1*)
  - Again recall: rotation formulas are *about the origin*
    - What *is* the result of applying *Rotate (θ)* to P2 ?



Transformations on Lines (cont.)

- Solution: as before – *force the rotation to be "about the origin"*



1. **P2.translate (-P1.x, -P1.y)**
2. **P2.rotate (θ)**
3. **P2.translate (P1.x, P1.y)**

Note "object-oriented" form

Transformations Using Matrices

- Translation

```
P = (x, y)

T = (+2, +3)

P′ = (x+2, y+3)
```

$$P' = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} 2 \\ 3 \end{bmatrix} = \begin{bmatrix} (x+2) \\ (y+3) \end{bmatrix}$$

Matrix Transformations (cont.)

- Rotation (CCW) about the origin

```
x′ = x cos(θ) - y sin(θ)
y′ = x sin(θ) + y cos (θ)
```

$$P' = \begin{bmatrix} x & y \end{bmatrix} * \begin{bmatrix} \cos(\theta) & \sin(\theta) \\ -\sin(\theta) & \cos(\theta) \end{bmatrix}$$

$$= \begin{bmatrix} (x\cos(\theta) - y\sin(\theta)) & (x\sin(\theta) + y\cos(\theta)) \end{bmatrix}$$

Matrix Transformations (cont.)

- Scaling

```
P = (x, y)

S = (sx, sy)

P′ = (x*sx, y*sy)
```

$$P' = \begin{bmatrix} x & y \end{bmatrix} * \begin{bmatrix} s_x & 0 \\ 0 & s_y \end{bmatrix}$$

$$= \begin{bmatrix} (x * s_x) & (y * s_y) \end{bmatrix}$$

Homogeneous Coordinates

- Motivation: uniformity between different matrix operations

- General Plan:
  - Represent a 2D point as a *triple* : **`[ x y 1 ]`**
  - Represent every transformation as a *3 x 3 matrix*
  - Use matrix ***multiplication*** for ***all*** transformations

Homogeneous Transformations

$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix}$$

Translation

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

Rotation

$$\begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

Scaling

Applying Transformations

- Translation

$$\begin{bmatrix} x & y & 1 \end{bmatrix} * \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ T_x & T_y & 1 \end{bmatrix} = \begin{bmatrix} (x + T_x) & (y + T_y) & 1 \end{bmatrix}$$

Applying Transformations (cont.)

- Rotation

$$[x \quad y \quad 1] * \begin{bmatrix} \cos(\theta) & \sin(\theta) & 0 \\ -\sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \left[\, (x\cos(\theta) - y\sin(\theta)) \quad (x\sin(\theta) + y\cos(\theta)) \quad 1 \,\right]$$

Applying Transformations (cont.)

- Scaling

$$\begin{bmatrix} x & y & 1 \end{bmatrix} * \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} (x * S_x) & (y * S_y) & 1 \end{bmatrix}$$

Column-Major Representation

- Translation:

$$\begin{bmatrix} (x+T_x) \\ (y+T_y) \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Rotation:

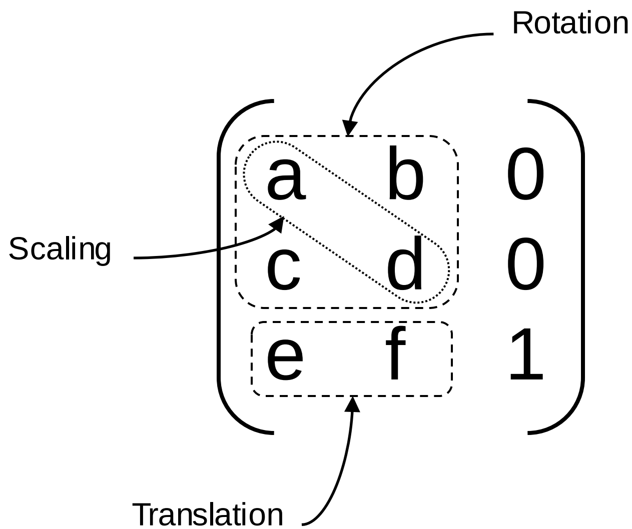
$$\begin{bmatrix} (x\cos(\theta) - y\sin(\theta)) \\ (x\sin(\theta) + y\cos(\theta)) \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\theta) & -\sin(\theta) & 0 \\ \sin(\theta) & \cos(\theta) & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

- Scaling:

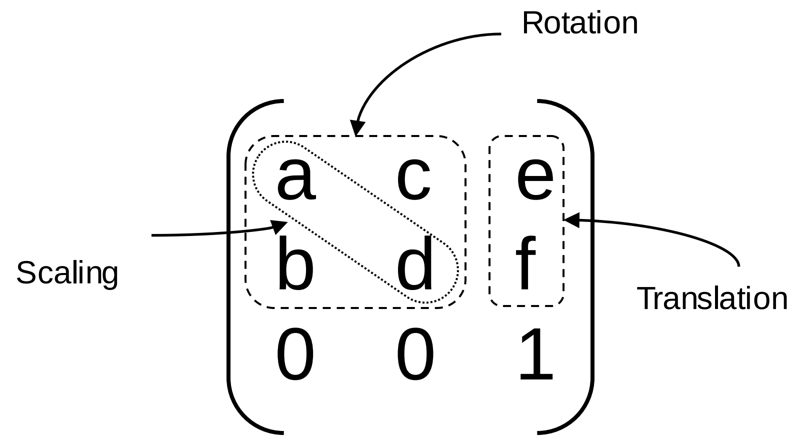
$$\begin{bmatrix} (x*S_x) \\ (y*S_y) \\ 1 \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Active Matrix Areas

Row-major form



Column-major form



Same size “active area” – 6 elements (3x2 or 2x3)

. Technically, these two ways of expressing points and vectors as matrices are perfectly valid and choosing one mode or the other is just a matter of convention.

Vector written as [1x3] matrix: $V = [x \quad y \quad z]$

Vector written as [3x1] matrix: $V = \begin{bmatrix} x \\ y \\ z \end{bmatrix}$

Source: https://www.scratchapixel.com/lessons/mathematics-physics-for-computer-graphics/geometry/row-major-vs-column-major-vector

Concatenation of Transforms

Typical Sequence:

```
P1 × Translate(tx,ty) = P2 ;
P2 × Rotate(θ) = P3 ;
P3 × Scale(sx,sy) = P4 ;
P4 × Translate(tx,ty) = P5 ;
```

Concatenation of Transforms (cont.)

- In (row-major) Matrix Form:

$$\begin{bmatrix} x1 & y1 & 1 \end{bmatrix} \times \begin{pmatrix} \text{Translate} \\ (tx, ty) \end{pmatrix} = \begin{bmatrix} x2 & y2 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x2 & y2 & 1 \end{bmatrix} \times \begin{pmatrix} \text{Rotate}(\theta) \end{pmatrix} = \begin{bmatrix} x3 & y3 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x3 & y3 & 1 \end{bmatrix} \times \begin{pmatrix} \text{Scale} \\ (sx, sy) \end{pmatrix} = \begin{bmatrix} x4 & y4 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x4 & y4 & 1 \end{bmatrix} \times \begin{pmatrix} \text{Translate} \\ (tx, ty) \end{pmatrix} = \begin{bmatrix} x5 & y5 & 1 \end{bmatrix}$$

Concatenation of Transforms (cont.)

- Alternate Matrix Form:

$$\left(\left(\left(\left[\begin{matrix} \mathbf{x1} & \mathbf{y1} & \mathbf{1} \end{matrix}\right] \times \left[\ \mathbf{T1}\ \right]\right) \times \left[\ \mathbf{R1}\ \right]\right) \times \left[\ \mathbf{S1}\ \right]\right) \times \left[\ \mathbf{T2}\ \right]\right)$$

$$= \left[\begin{matrix} \mathbf{x5} & \mathbf{y5} & \mathbf{1} \end{matrix}\right]$$

Concatenation of Transforms (cont.)

- Matrix multiplication is *associative* :

$$\begin{bmatrix} x1 & y1 & 1 \end{bmatrix} \times \underbrace{\left(\begin{bmatrix} T1 \end{bmatrix} \times \begin{bmatrix} R1 \end{bmatrix} \times \begin{bmatrix} S1 \end{bmatrix} \times \begin{bmatrix} T2 \end{bmatrix} \right)} = \begin{bmatrix} x5 & y5 & 1 \end{bmatrix}$$

$$\begin{bmatrix} x1 & y1 & 1 \end{bmatrix} \times \begin{bmatrix} M \end{bmatrix} = \begin{bmatrix} x5 & y5 & 1 \end{bmatrix}$$

In Column-Major Form

$$\begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix} = \begin{bmatrix} \text{Trans} \\ (x, y) \end{bmatrix} \times \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix} = \begin{bmatrix} Rot(\theta) \end{bmatrix} \times \begin{bmatrix} x_2 \\ y_2 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_4 \\ y_4 \\ 1 \end{bmatrix} = \begin{bmatrix} \text{Scale} \\ (sx, sy) \end{bmatrix} \times \begin{bmatrix} x_3 \\ y_3 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_5 \\ y_5 \\ 1 \end{bmatrix} = \begin{bmatrix} \text{Trans} \\ (x, y) \end{bmatrix} \times \begin{bmatrix} x_4 \\ y_4 \\ 1 \end{bmatrix}$$

Column-Major Form (cont.)

$$\begin{bmatrix} x_5 \\ y_5 \\ 1 \end{bmatrix} = \left(\begin{bmatrix} T2 \end{bmatrix} \times \left(\begin{bmatrix} S1 \end{bmatrix} \times \left(\begin{bmatrix} R1 \end{bmatrix} \times \left(\begin{bmatrix} T1 \end{bmatrix} \times \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix} \right) \right) \right) \right)$$

$$\begin{bmatrix} x_5 \\ y_5 \\ 1 \end{bmatrix} = \underbrace{\left(\begin{bmatrix} T2 \end{bmatrix} \times \begin{bmatrix} S1 \end{bmatrix} \times \begin{bmatrix} R1 \end{bmatrix} \times \begin{bmatrix} T1 \end{bmatrix} \right)} \times \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$

$$\begin{bmatrix} x_5 \\ y_5 \\ 1 \end{bmatrix} = \begin{bmatrix} M \end{bmatrix} \times \begin{bmatrix} x_1 \\ y_1 \\ 1 \end{bmatrix}$$