2D Transformations

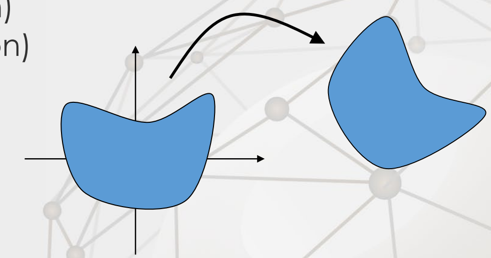
Learning objectives

By the end of the module, you should be able to:

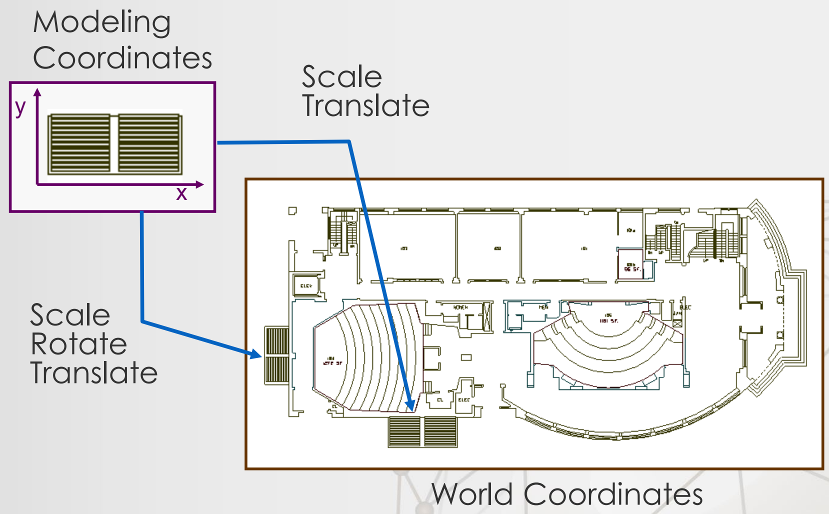
- Identify and explain basic 2D transformations
- Perform conversion between Cartesian coordinates and homogeneous coordinates
- Understand and explain affine transformations
- Represent and construct affine transformations by matrix or matrices
- Perform computation of 2D transformations using matrices and vectors

1. Introduction

- Transformations are the lifeblood of geometry!
- In computer graphics, transformations are used to position, orient, and scale objects as well as to model shape.
- For example, given a 2D object, transformation can be used to change the object's:
  - Position (translation)
  - Orientation (rotation)
  - Size (scaling)
  - Shape (shear)



Example: 2D transformations



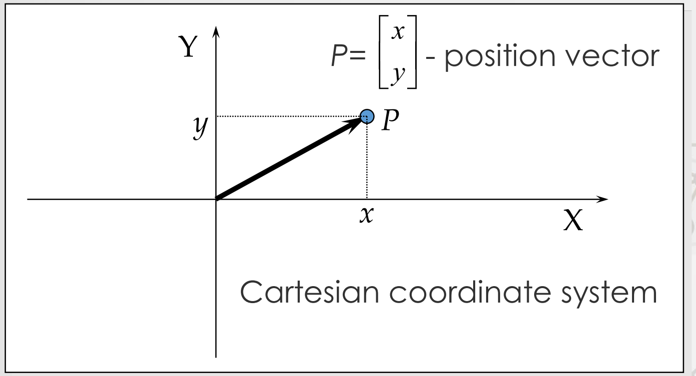
Problems to be addressed

- What are basic 2D transformations?
- What are homogeneous coordinates?
- What are 2D affine transformations?
- How to represent 2D transformations using matrix/ matrices?
- How to perform 2D transformation?

2. Basic 2D transformations

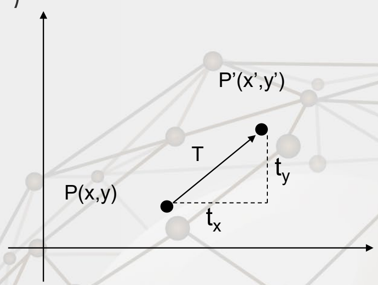
- Basic set:
  - Translation
  - Scaling
  - Rotation
- Some other simple transformations:
  - Reflection
  - Shear

Point representation



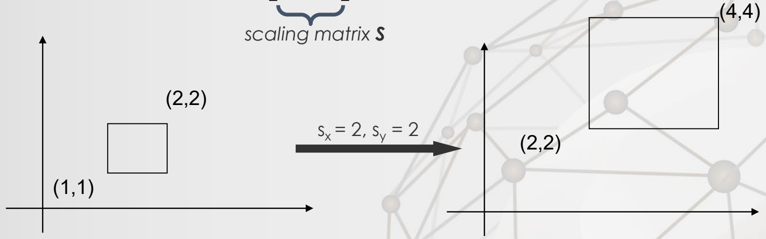
Translation

- Moves a point to a new location by adding translation amounts to the coordinates of the point.
- P(x,y) $\xrightarrow{\text{Translate by } T=(t_x,t_y)}$ P'(x',y')
- How to compute P'?
- $x' = x+t_x,\ y' = y+t_y$
- That is, $\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \end{bmatrix}$ or ***P' = P+T.***



Scaling

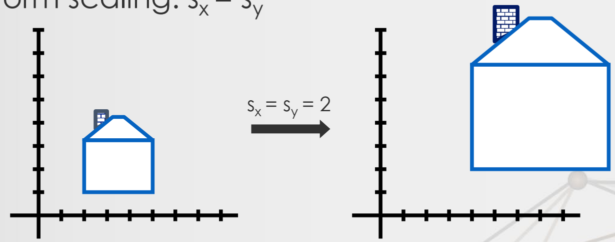
- Changes the size of the object by multiplying the coordinates of the points by scaling factors (s_x, s_y).
- P = (x, y) $\xrightarrow{\text{scaling}}$ P' = (x', y')
- How to compute P'?
- $\begin{bmatrix}x'\\y'\end{bmatrix}=\begin{bmatrix}s_x x\\s_y y\end{bmatrix}$ or $\begin{bmatrix}x'\\y'\end{bmatrix}=\underbrace{\begin{bmatrix}s_x & 0\\0 & s_y\end{bmatrix}}_{\text{scaling matrix } \boldsymbol{S}}\begin{bmatrix}x\\y\end{bmatrix}$ or ***P' = SP***.



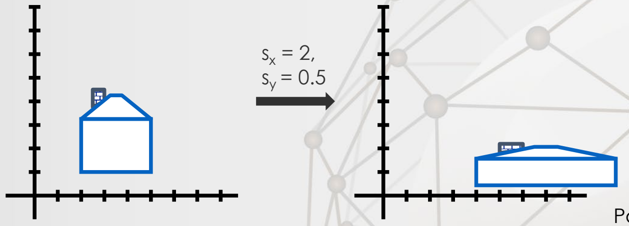
Note: When the size is changed, the object may also move.

Uniform vs non-uniform scaling

- Uniform scaling: $s_x = s_y$

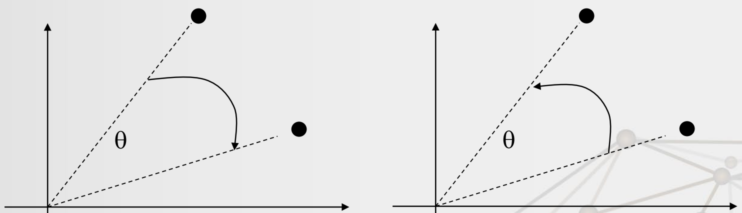


- Non-uniform scaling: $s_x \neq s_y$



Rotation

- Rotates a point about origin (0, 0) by an angle θ



θ < 0: Rotate clockwise

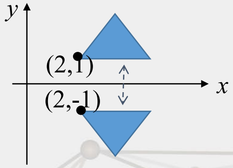
θ > 0: Rotate counter clockwise

- P = (x, y) $\xrightarrow{\text{rotate about origin by } \theta}$ P' = (x', y')
- How to compute (x', y')?

Other 2D transformations: Reflection

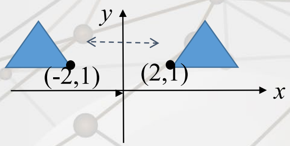
- Produces a mirror image of an object.
  - Reflection about the x-axis:

    $x' = x$, $y' = -y$ or $\begin{bmatrix}x'\\y'\end{bmatrix}=\begin{bmatrix}1 & 0\\0 & -1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$



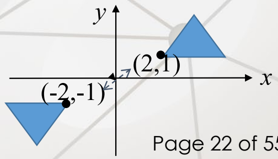
  - Reflection about the y-axis:

    $x' = -x$, $y' = y$ or $\begin{bmatrix}x'\\y'\end{bmatrix}=\begin{bmatrix}-1 & 0\\0 & 1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$



  - Reflection relative to the coordinate origin:

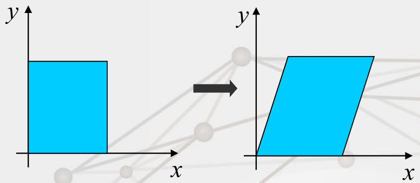
    $x' = -x$, $y' = -y$ or $\begin{bmatrix}x'\\y'\end{bmatrix}=\begin{bmatrix}-1 & 0\\0 & -1\end{bmatrix}\begin{bmatrix}x\\y\end{bmatrix}$



Other 2D transformations: Shear

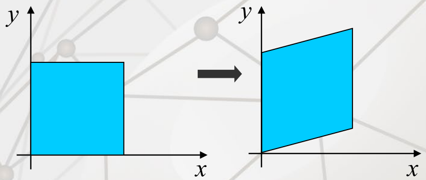
- Changes the shape of an object such that the transformed shape appears as if the object were composed of internal layers that had been caused to slide over each other.
  - x-direction shear:

$$x' = x + a \cdot y \qquad y' = y \quad \text{or} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & a \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



  - y-direction shear:

$$x' = x \qquad y' = b \cdot x + y \quad \text{or} \quad \begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ b & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$



Recap

- Translation: P' = P + T
- Scaling: P' = SP
- Rotation: P' = RP
- Reflection: P' = R_xP, P' = R_yP, P' = R_oP
- Shear: P' = H_xP, P' = H_yP

where S, R, R_x, R_y, R_o, H_x, H_y are 2×2 matrices.

Homogeneous coordinates

- If 2D Cartesian coordinate representation (x,y) is expanded to a three-element representation (x_h, y_h, h) where h is a non-zero value such that

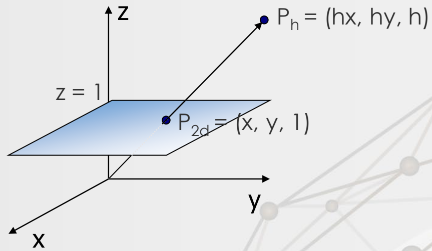
$$x = \frac{x_h}{h}, \quad y = \frac{y_h}{h},$$

then (x_h, y_h, h) is called **homogeneous coordinates** of point (x, y).

- (x, y) has multiple representations in homogeneous coordinates. For example, x_h = hx, y_h = hy.
  - h = 1 (x, y) → (x, y, 1)
  - h = 2 (x, y) → (2x, 2y, 2)

Geometric meaning

- (x, y) ↔ P_{2d} = (x, y, 1)
- P_{2d} is a **projection** of P_h=(hx, hy, h) onto the z = 1 plane.



- An infinite number of points correspond to P_{2d} . They constitute the whole line (hx, hy, h).
- For example, (2, 1, 1), (4, 2, 2), (-6, -3, -3) all represent point (2, 1).

Representation conversion

- Cartesian → Homogeneous

$$\begin{bmatrix} x \\ y \end{bmatrix} \xrightarrow{\text{homogenizing}} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = \begin{bmatrix} hx \\ hy \\ h \end{bmatrix}, h \neq 0$$

- Homogeneous → Cartesian

$$\begin{bmatrix} x_h \\ y_h \\ h \end{bmatrix} = \begin{bmatrix} \frac{x_h}{h} \\ \frac{y_h}{h} \\ 1 \end{bmatrix} \xrightarrow{\text{inhomogenizing}} \begin{bmatrix} \frac{x_h}{h} \\ \frac{y_h}{h} \end{bmatrix}$$

Examples: Homogeneous vs Cartesian

Q: Which of the following points defined using homogeneous coordinates are identical to the point with Cartesian coordinates (1, 2)?

$$\underset{\text{(i)}}{\begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix}} \quad \underset{\text{(ii)}}{\begin{bmatrix} 1 \\ 2 \\ 2 \end{bmatrix}} \quad \underset{\text{(iii)}}{\begin{bmatrix} 2 \\ 4 \\ 1 \end{bmatrix}} \quad \underset{\text{(iv)}}{\begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix}} \quad \underset{\text{(v)}}{\begin{bmatrix} -1 \\ -2 \\ 1 \end{bmatrix}} \quad \underset{\text{(vi)}}{\begin{bmatrix} -3 \\ -6 \\ -3 \end{bmatrix}} \quad \underset{\text{(vii)}}{\begin{bmatrix} 1 \\ 2 \\ -1 \end{bmatrix}} \quad \underset{\text{(viii)}}{\begin{bmatrix} 1.5 \\ 3 \\ 1.5 \end{bmatrix}} \quad \underset{\text{(ix)}}{\begin{bmatrix} -1 \\ -2 \\ -1 \end{bmatrix}}$$

Hints: The main idea is to divide the vector by its 3rd component to make the 3rd component be 1, and then to extract the first two components. For example, consider (iv).

$$\begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} \rightarrow \begin{bmatrix} 2 \\ 4 \\ 2 \end{bmatrix} /2 = \begin{bmatrix} 1 \\ 2 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 1 \\ 2 \end{bmatrix}$$

Ans: (i), (iv), (vi), (viii), (ix).

Advantages of homogeneous coordinates

- Homogeneous coordinates seem to be unintuitive, but they make graphics operations easier (in hardware and software).
  - In particular, homogeneous coordinates allow all three transformations (translation, rotation and scaling) to be expressed using 3×3 matrices, which makes transformation composition be expressed as multiplication of matrices.

4. Representation using 3x3 matrix

- With homogeneous coordinates, all basic 2D transformations can be represented using 3x3 matrices.
- Matrices are a convenient and efficient way to represent a sequence of transformations so that we can perform all transformations using matrix/vector multiplication.
  - General purpose representation, and
  - Hardware matrix multiplication.

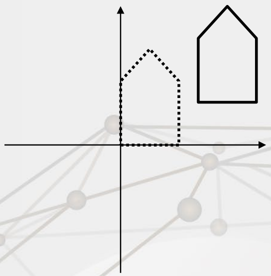
2D translation

- 2D Cartesian coordinate representation

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

- Homogeneous representation

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



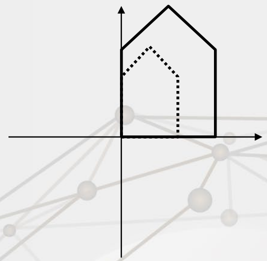
2D scaling

- 2D Cartesian coordinate representation

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

- Homogeneous representation

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



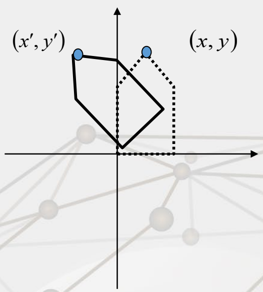
2D rotation

- 2D Cartesian coordinate representation

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

- Homogeneous representation

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



2D reflections

Cartesian coordinate $\longrightarrow$ Homogeneous coordinates

- About x-axis:

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

- About y-axis:

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

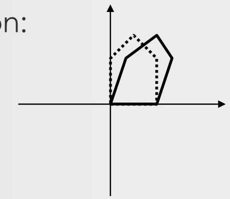
- Over the origin:

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

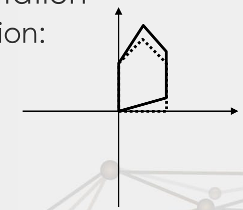
2D shear

- 2D Cartesian coordinate representation

x-direction:



y-direction:



- Homogeneous representation

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & a \\ 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix} \qquad \begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{pmatrix} 1 & 0 \\ b & 1 \end{pmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & a & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix} \qquad \begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ b & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

Recap

- Translation: $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$
- Scaling: $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$
- Rotation: $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$
- Reflection: $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$ $\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$

2D affine transformations

- We will first look at how basic 2D transformations are composed to form one transformation.
- Then, we will look at affine transformations that are generalizations of basic transformations.

Composing transformation

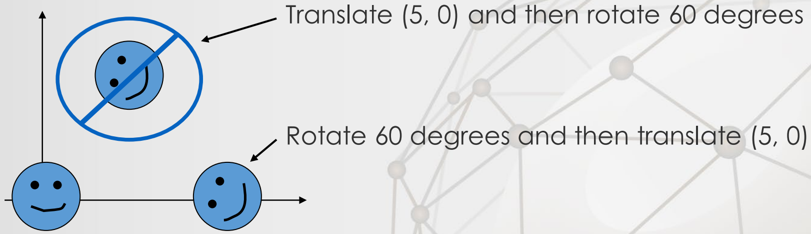
- Composing transformations – the process of applying several transformations in succession to form one overall transformation.
- If we apply transformation to a point P using matrix M_1 first, and then transformations using M_2, and M_3, then we have:

$$(M_3 \times (M_2 \times (M_1 \times P))) = \underbrace{M_3 \times M_2 \times M_1}_{M} \times P$$

(pre-multiply)

Transformation order

- Matrix multiplication is **associative**:
  $M_3 \times M_2 \times M_1 = (M_3 \times M_2) \times M_1 = M_3 \times (M_2 \times M_1)$
- Matrix multiplication may not be commutative:
  $A \times B \neq B \times A$.
- Example: Rotation and translation are not commutative.



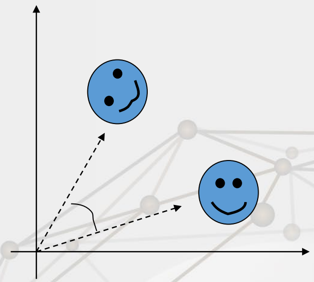
Applications of composing transformation

- What happens if you rotate an object about an arbitrary point (not the origin)?
- What should you do to resize an object about its own center?

Rotation revisit

- The standard rotation matrix is used to rotate about the origin (0, 0).

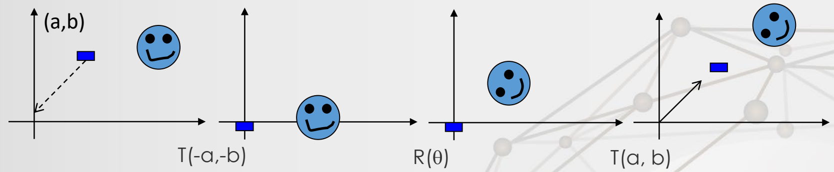
$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



- What if I want to rotate about an arbitrary center?

Rotation about an arbitrary point

- To rotate about an arbitrary point P=(a, b) by θ:
  - Translate the object so that P will coincide with the origin: T(-a, -b);
  - Rotate the object: R(θ);
  - Translate the object back: T(a, b).



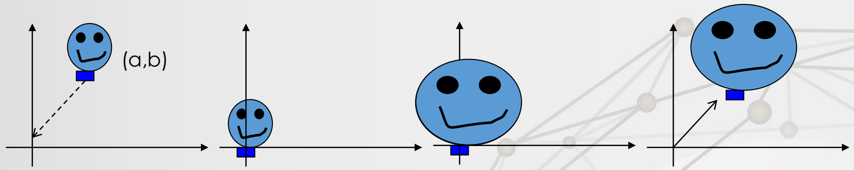
- Thus, P' = T(a, b) R(θ) T(-a, -b) P.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Scaling with regards to an arbitrary pivot

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

- To scale about an arbitrary point P=(a, b) by (s_x, s_y):
  - Translate the object so that P will coincide with the origin: T(-a, -b);
  - Scale the object: S(s_x, s_y);
  - Translate the object back: T(a, b).



- Thus, P' = T(a, b) S(s_x, s_y) T(-a, -b) P.

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \left(\begin{bmatrix} 1 & 0 & a \\ 0 & 1 & b \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -a \\ 0 & 1 & -b \\ 0 & 0 & 1 \end{bmatrix} \right) \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Example 1

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

Q: A 2D geometric object is rotated about the origin by 90° in clockwise direction, and then scaled relative to the point (3, 5) in the x-direction by 5 times and in the y-direction by 2 times. Finally, the object is reflected through the point (3, 5). Write in a proper order the matrices composing this transformation.

Hint: Basic approach –

Rotate:

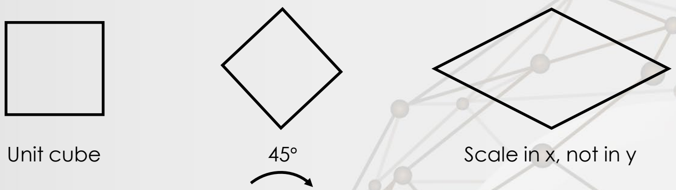
Scale: translate to the origin, scale, translate back.

Reflect: translate to the origin, reflect, translate back.

Affine transformation

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

- Affine transformations map straight lines to straight lines and preserve ratios of distances along straight lines.
- Affine transformations preserve parallelism of lines but not lengths and angles.



- Affine transformations are composites of four transformations: translation, rotation, scaling, and shear.

Matrix form of affine transformation

NANYANG TECHNOLOGICAL UNIVERSITY SINGAPORE

- Affine transformations can always be represented by:

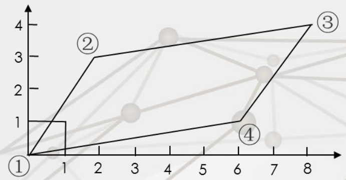
$$x' = ax + by + m$$
$$y' = cx + dy + n$$

  where
  - a, b, c, d, m, n are constants;
  - (x, y) are the coordinates of the point to be transformed;
  - (x', y') are the coordinates of the transformed point.
- The general matrix form of affine transformations is:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Example 2

Q: Refer to the figure. A unit square is transformed by an affine transformation such that the 4 corners of the square with coordinates (0, 0), (0, 1), (1, 1), (1, 0) are mapped to vertices labeled by ①,②,③,④, respectively. Find the matrix of this affine transformation.



Ans:

From the figure, we find that vertices ①,②,③,④ have coordinates (0, 0), (2, 3), (8, 4), and (6, 1).

Let the transformation matrix be $[T]=\begin{bmatrix} a & b & m \\ c & d & n \\ 0 & 0 & 1 \end{bmatrix}$.

Then, [P'] = [T][P].

That is, **x' = ax + by + m**, **y' = cx + dy + n**.

6. Summary

- 2D point and homogeneous coordinates
- Basic 2D transformations and their 3x3 matrix representation
- Composition of transformations
- 2D affine transformations and their 3x3 matrix representation