FLAME UNIVERSITY

EVERLASTING learning

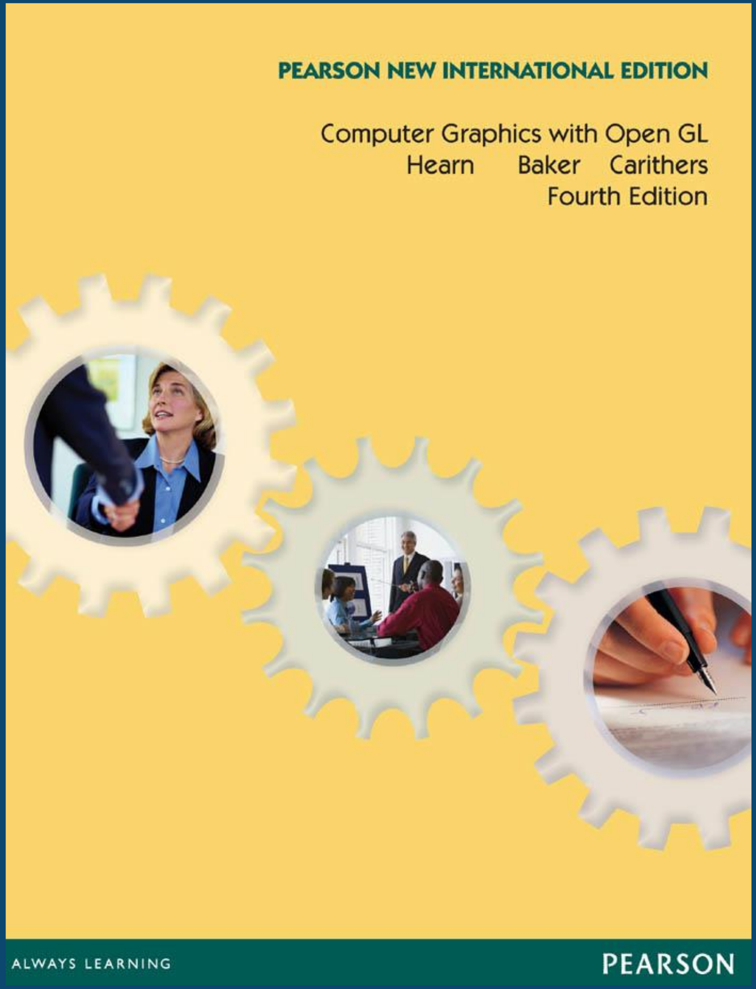
FUNDAMENTALS OF COMPUTER GRAPHICS (CSIT304)

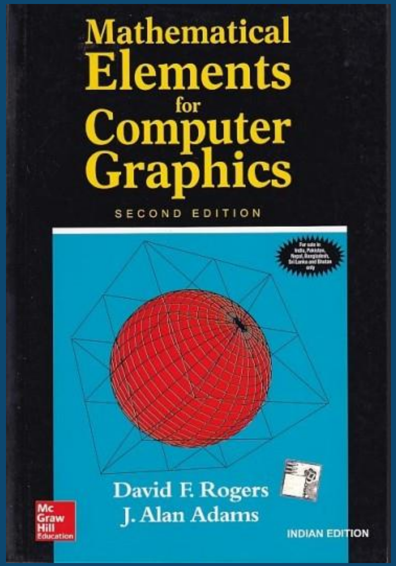
RASTERIZATION AND 2D TRANSFORMATION

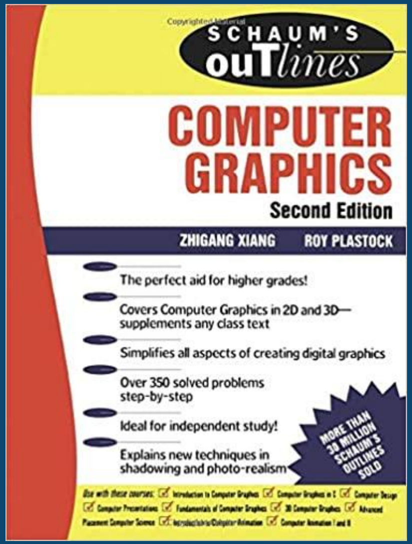
CHIRANJOY CHATTOPADHYAY

Associate Professor,

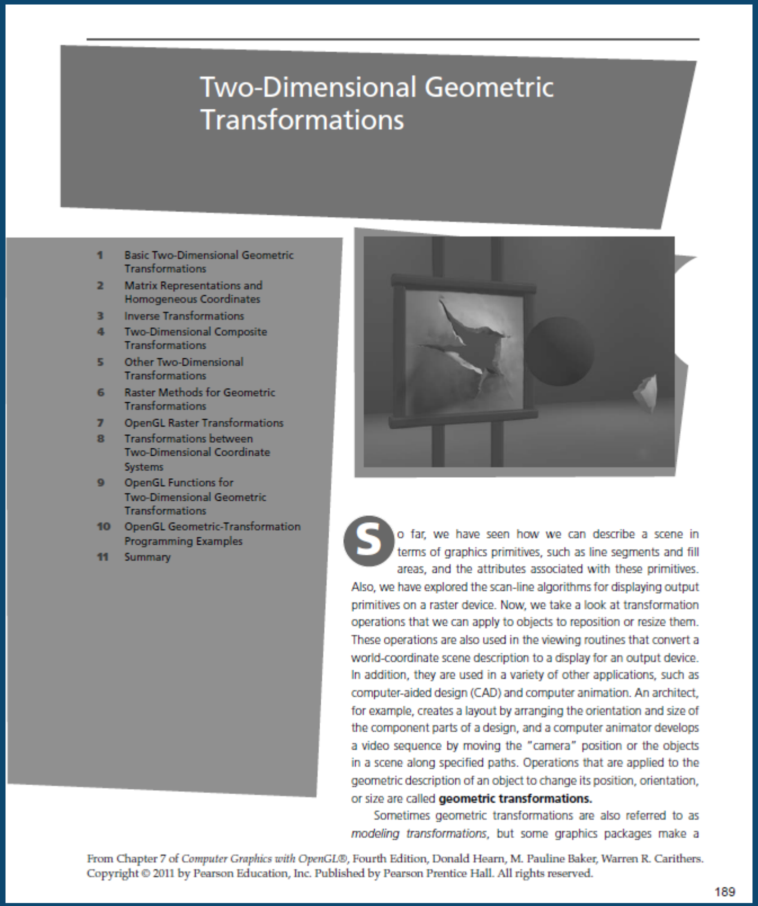
FLAME School of Computation and Data Science







2D TRANSFORMATION



Two-Dimensional Geometric Transformations

1 Basic Two-Dimensional Geometric Transformations
2 Matrix Representations and Homogeneous Coordinates
3 Inverse Transformations
4 Two-Dimensional Composite Transformations
5 Other Two-Dimensional Transformations
6 Raster Methods for Geometric Transformations
7 OpenGL Raster Transformations
8 Transformations between Two-Dimensional Coordinate Systems
9 OpenGL Functions for Two-Dimensional Geometric Transformations
10 OpenGL Geometric-Transformation Programming Examples
11 Summary

So far, we have seen how we can describe a scene in terms of graphics primitives, such as line segments and fill areas, and the attributes associated with these primitives. Also, we have explored the scan-line algorithms for displaying output primitives on a raster device. Now, we take a look at transformation operations that we can apply to objects to reposition or resize them. These operations are also used in the viewing routines that convert a world-coordinate scene description to a display for an output device. In addition, they are used in a variety of other applications, such as computer-aided design (CAD) and computer animation. An architect, for example, creates a layout by arranging the orientation and size of the component parts of a design, and a computer animator develops a video sequence by moving the "camera" position or the objects in a scene along specified paths. Operations that are applied to the geometric description of an object to change its position, orientation, or size are called **geometric transformations.**

Sometimes geometric transformations are also referred to as *modeling transformations*, but some graphics packages make a

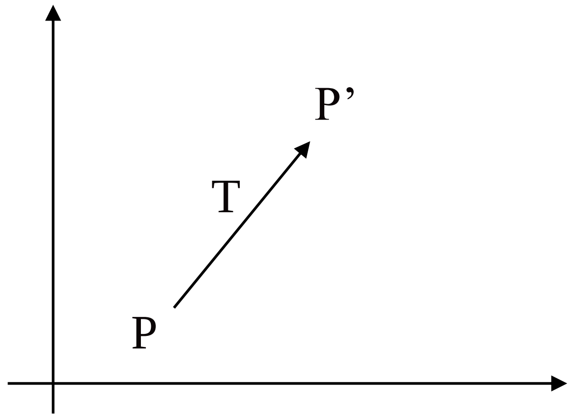
189

INTRODUCTION

- Sometimes also called modeling transformations
- Geometric transformations
  - Changing an object's position (translation), orientation (rotation) or size (scaling)
- Modeling transformations
  - Constructing a scene or hierarchical description of a complex object
- Others transformations: reflection and shearing operations

2D TRANSFORMATION: TRANSLATION

- 2D Translation



- $x' = x + t_x$, $y' = y + t_y$

$$P = \begin{bmatrix} x \\ y \end{bmatrix}, P' = \begin{bmatrix} x' \\ y' \end{bmatrix}, T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

- P'=P+T

- Translation moves the object without deformation (rigid-body transformation)

2D TRANSLATION

- To move a line segment

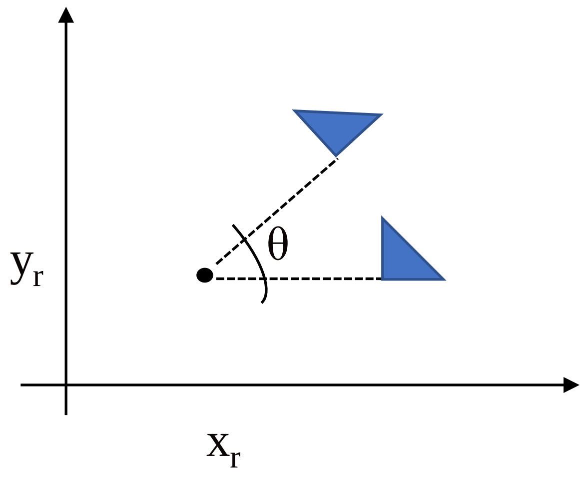
  1. Apply the transformation equation to each of the endpoints
  2. Redraw the line between new endpoints

- To move a polygon

  1. Apply the transformation equation to coordinates of each vertex
  2. Regenerate the polygon using the new set of vertex coordinates

ROTATION

- 2D Rotation
  - Rotation axis
  - Rotation angle
  - Rotation point or pivot point (xr,yr)

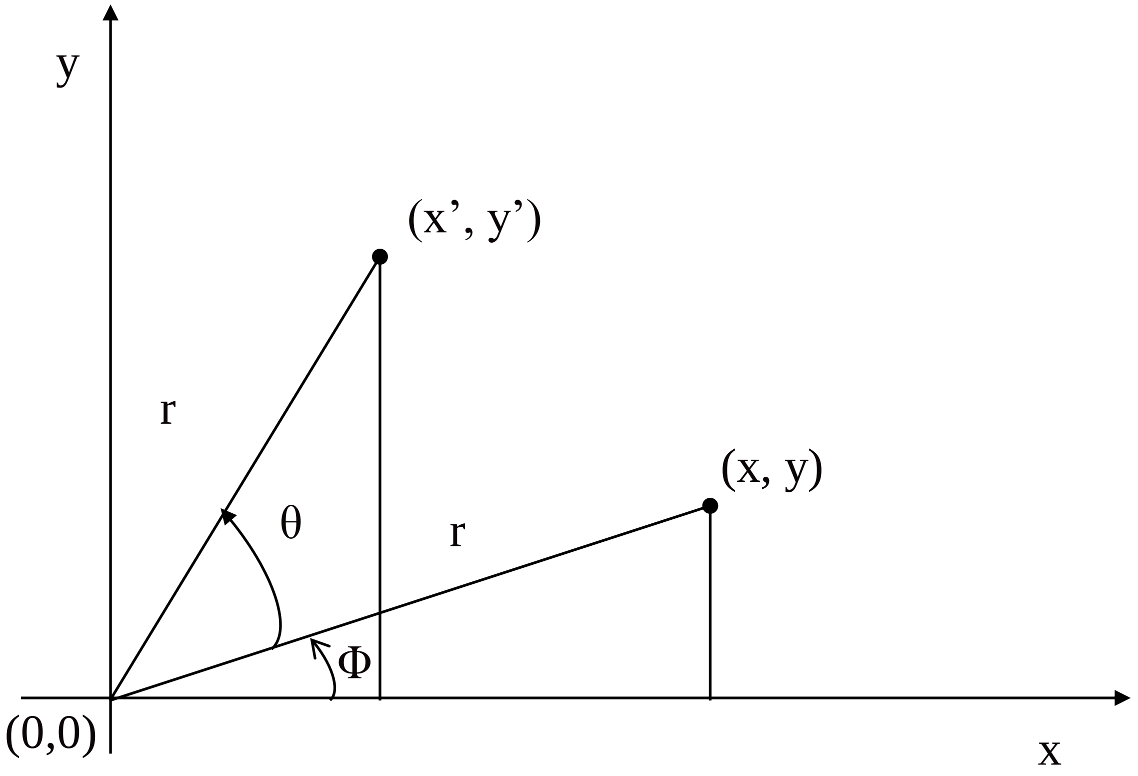


ROTATION

- If θ is positive
  - Counter-clockwise rotation
- If θ is negative
  - clockwise rotation
- Remember:
  - $\cos(a + b) = \cos a \cos b - \sin a \sin b$
  - $\cos(a - b) = \cos a \sin b + \sin a \cos b$

ROTATION FORMULA

- Suppose the pivot point is at the origin
- $x' = r\cos(\theta + \Phi)$
    - $= r\cos\theta\cos\Phi - r\sin\theta\sin\Phi$
- $y' = r\sin(\theta + \Phi)$
    - $= r\cos\theta\sin\Phi + r\sin\theta\cos\Phi$
- $x = r\cos\Phi,\ y = r\sin\Phi$
- **$x' = x\cos\theta - y\sin\theta$**
- **$y' = x\sin\theta + y\cos\theta$**



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

ROTATION

- Rotation about an arbitrary point?
- Move objects without deformation
- A line is rotated by
    1. Applying the rotation formula to each of the endpoints
    2. Redrawing the line between the new end points
- A polygon is rotated by
    1. Applying the rotation formula to each of the vertices
    2. Redrawing the polygon using new vertex coordinates

SCALING

- Scaling is used to **alter the size of an object**
- Multiply object positions (x, y) by scaling factors sx and sy
  - x′ = x · sx
  - y′ = y · sx

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

SCALING

- Any positive value can be used as scaling factor
  - Values less than 1 reduce the size of the object
  - Values greater than 1 enlarge the object
  - If scaling factor is 1 then the object stays unchanged
- If sx = sy , we call it **uniform scaling**
- If scaling factor <1, then
  - the object moves closer to the origin
- If scaling factor >1,
  - then the object moves farther from the origin

Why?

SCALING

- We can control the location of the scaled object by choosing a position called the **fixed point (xf, yf)**
- $x' - xf = (x - xf)\, sx \qquad y' - yf = (y - yf)\, sy$
- $x' = x \cdot sx + xf\,(1 - sx)$
- $y' = y \cdot sy + yf\,(1 - sy)$
- Polygons are scaled by
  1. Applying the above formula to each vertex
  2. Regenerating the polygon using the transformed vertices

COMBINING TRANSFORMATIONS

- We have a general transformation of a point:
  - $P' = M \cdot P + A$

- Is it possible to use the same matrix operation all the time?

- How to combine multiplication and addition into a single operation?

HOMOGENEOUS COORDINATES

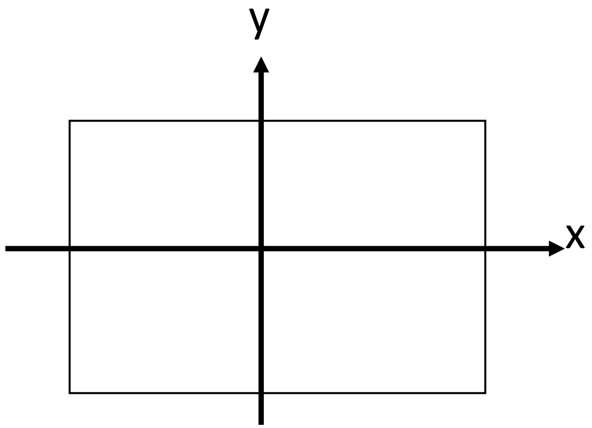
- Uniform representation of translation, rotation, scaling
- Uniform representation of points and vectors
- Compact representation of sequence of transformations

HOMOGENEOUS COORDINATE

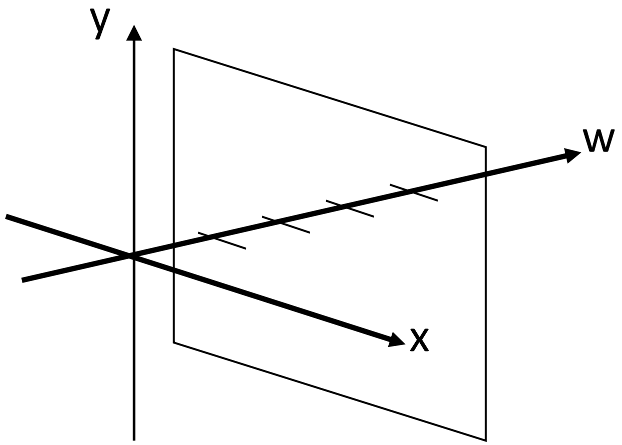
- Add extra coordinate:

  $\mathbf{P} = (p_x, p_y, p_h)$ or

  $\mathbf{x} = (x, y, h)$



➔



- Cartesian coordinates: divide by *h*

  $\mathbf{x} = (x/h, y/h)$

- Points: $h = 1$ (for the time being…),

HOMOGENEOUS COORDINATE

- We can always map back to the original 2D point by dividing by the last coordinate
- (15, 6, 3) ---→ (5, 2).

- Why do we use 1 for the last coordinate?

- The fact that all the points along each line can be mapped back to the same point in 2D gives this coordinate system its name – **homogeneous coordinates.**

MATRIX REPRESENTATION

- Point in column-vector:

$$\begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

- A point now has three coordinates.
- So the matrix is needs to be 3x3.
- Translation

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

MATRIX REPRESENTATION

- **Rotation**

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

- **Scaling**

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

COMPOSITE TRANSFORMATION

- We can represent any sequence of transformations as a single matrix.
    - No special cases when transforming a point – matrix • vector.
    - Composite transformations – matrix • matrix.

- Composite transformations:
    - Rotate about an arbitrary point – translate, rotate, translate
    - Scale about an arbitrary point – translate, scale, translate
    - Change coordinate systems – translate, rotate, scale

- **Does the order of operations matter?**

COMPOSITION PROPERTIES

- Is matrix multiplication associative?
  - (A.B).C = A.(B.C)

$$\left(\begin{bmatrix} a & b \\ c & d \end{bmatrix} \bullet \begin{bmatrix} e & f \\ g & h \end{bmatrix}\right) \bullet \begin{bmatrix} i & j \\ k & l \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix} \bullet \begin{bmatrix} i & j \\ k & l \end{bmatrix}$$

$$= \begin{bmatrix} aei+bgi+afk+bhk & aej+bgj+afl+bhl \\ cei+dgi+cfk+dhk & cej+dgj+cfl+dhl \end{bmatrix}$$

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \bullet \left(\begin{bmatrix} e & f \\ g & h \end{bmatrix} \bullet \begin{bmatrix} i & j \\ k & l \end{bmatrix}\right) = \begin{bmatrix} a & b \\ c & d \end{bmatrix} \bullet \begin{bmatrix} ei+fk & ej+fl \\ gi+hk & gj+hl \end{bmatrix}$$

$$= \begin{bmatrix} aei+afk+bgi+bhk & aej+afl+bgj+bhl \\ cei+cfk+dgi+dhk & cej+cfl+dgj+dhl \end{bmatrix}$$

COMPOSITION PROPERTIES

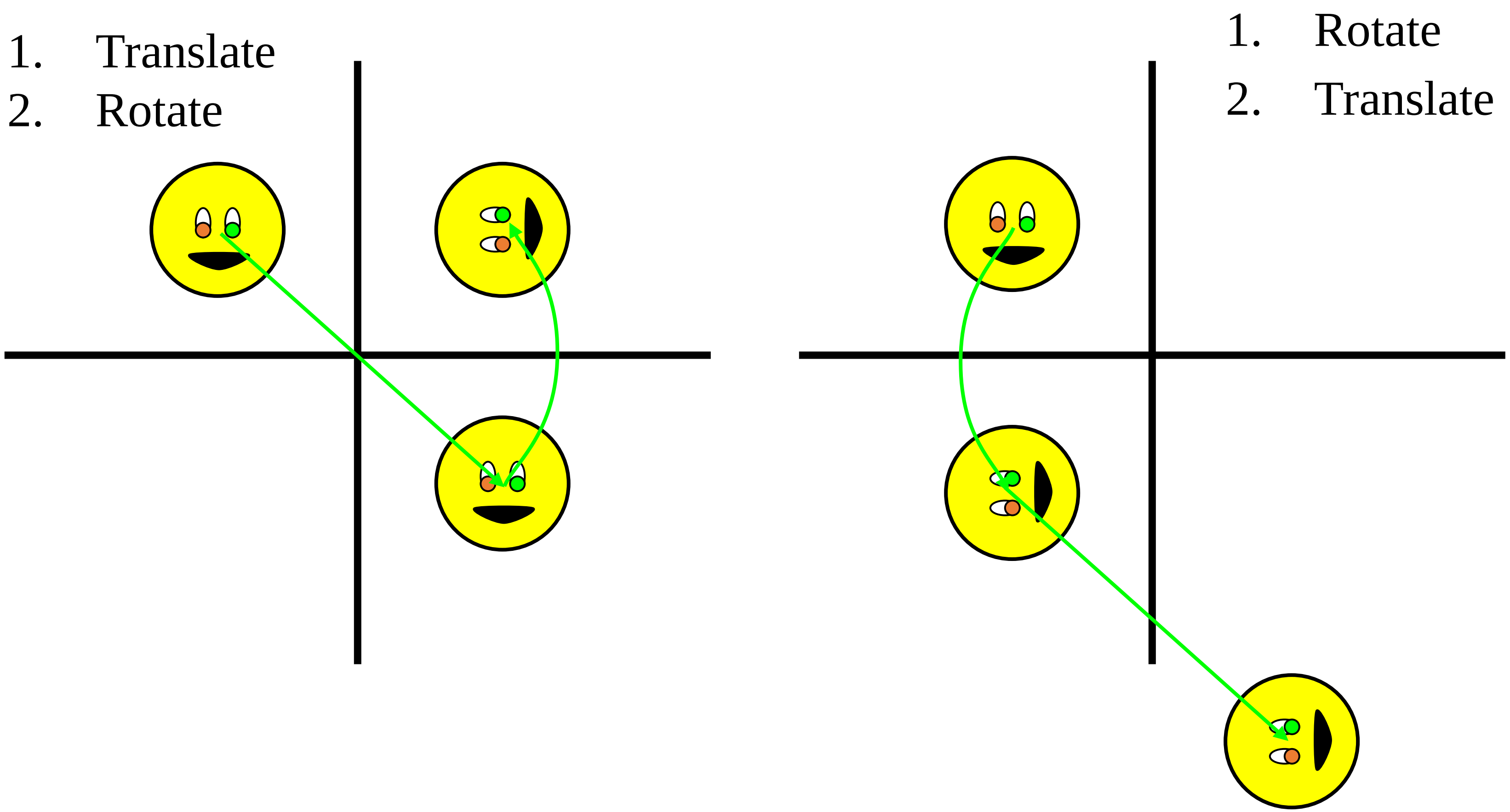
- Is matrix multiplication commutative?
  - A . B = B . A ?

$$\begin{bmatrix} a & b \\ c & d \end{bmatrix} \bullet \begin{bmatrix} e & f \\ g & h \end{bmatrix} = \begin{bmatrix} ae+bg & af+bh \\ ce+dg & cf+dh \end{bmatrix}$$

$$\begin{bmatrix} e & f \\ g & h \end{bmatrix} \bullet \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} ea+fc & eb+fd \\ ga+hc & gb+hd \end{bmatrix}$$

ORDER OF OPERATIONS

- So, it does matter. Let's look at an example:



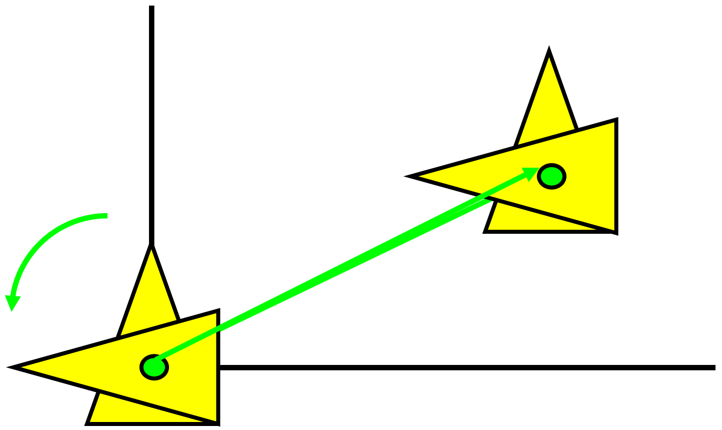
COMPOSITE TRANSFORMATION MATRIX

- Arrange the transformation matrices in order from right to left.
- General Pivot- Point Rotation
  - Operation :- T(pivot) • R(θ) • T(–pivot)
    1. Translate (pivot point is moved to origin)
    2. Rotate about origin
    3. Translate (pivot point is returned to original position)

$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{pmatrix}$$

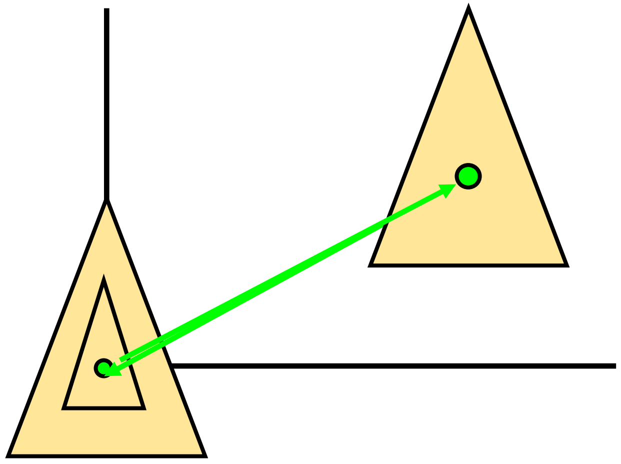
$$\begin{pmatrix} 1 & 0 & t_x \\ 0 & 1 & t_y \\ 0 & 0 & 1 \end{pmatrix} \cdot \begin{pmatrix} \cos\theta & -\sin\theta & -t_x\cos\theta + t_y\sin\theta \\ \sin\theta & \cos\theta & -t_x\sin\theta - t_y\cos\theta \\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos\theta & -\sin\theta & -t_x\cos\theta + t_y\sin\theta + t_x \\ \sin\theta & \cos\theta & -t_x\sin\theta - t_y\cos\theta + t_y \\ 0 & 0 & 1 \end{pmatrix}$$



COMPOSITE TRANSFORMATION MATRIX

- General Fixed-Point Scaling
- Operation :- T(fixed) • S(scale) • T(–fixed)
- Translate (fixed point is moved to origin)
- Scale with respect to origin
- Translate (fixed point is returned to original position)



Exercise:

1. **Find the matrix that represents scaling of an object with respect to any fixed point?**

2. **Given P(6, 8) , Sx = 2, Sy = 3 and fixed point (2, 2). Use that matrix to find P'?**

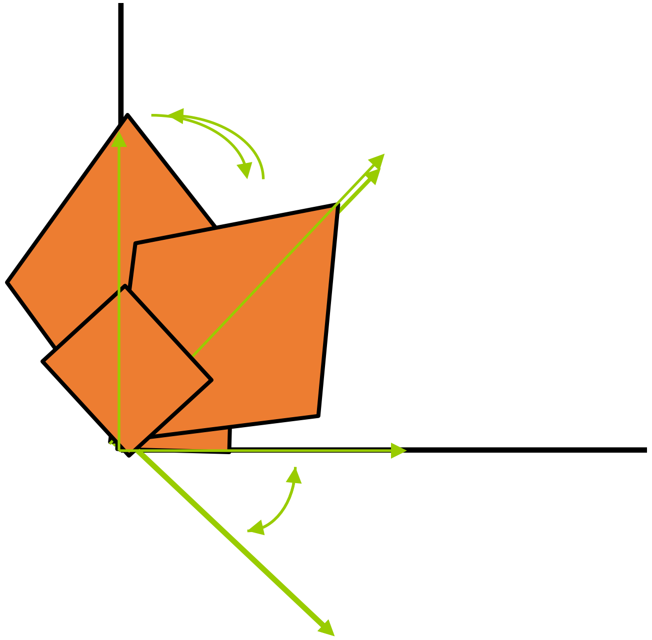
COMPOSITE TRANSFORMATION MATRIX

General Scaling Direction

Operation :-

1. Rotate (scaling direction align with the coordinate axes)
2. Scale with respect to origin
3. Rotate (scaling direction is returned to original position)

$R(-\theta) \bullet S(scale) \bullet R(\theta)$



Exercise:

1. Find the composite transformation matrix.

INVERSE TRANSFORMATIONS

$$\mathbf{T}^{-1} = \begin{bmatrix} 1 & 0 & -t_x \\ 0 & 1 & -t_y \\ 0 & 0 & 1 \end{bmatrix}$$

Translation

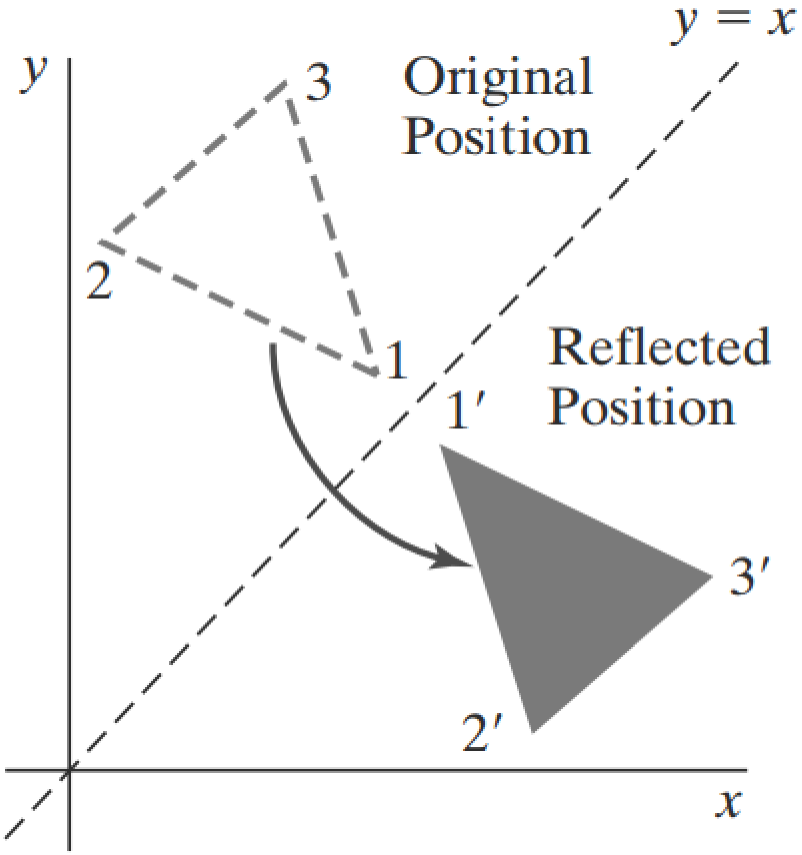
$$\mathbf{R}^{-1} = \begin{bmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

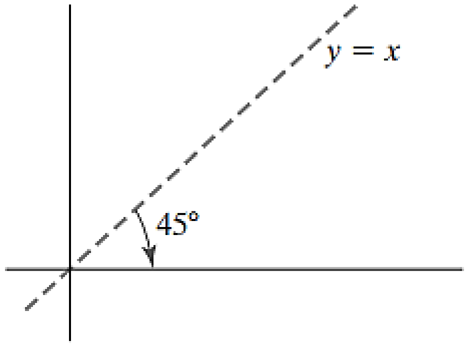
Rotation

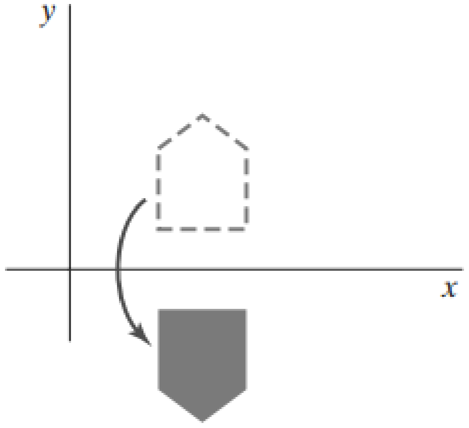
$$\mathbf{S}^{-1} = \begin{bmatrix} \frac{1}{s_x} & 0 & 0 \\ 0 & \frac{1}{s_y} & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

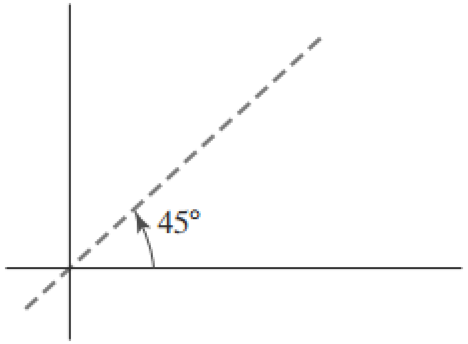
Scaling

REFLECTION ABOUT A LINE







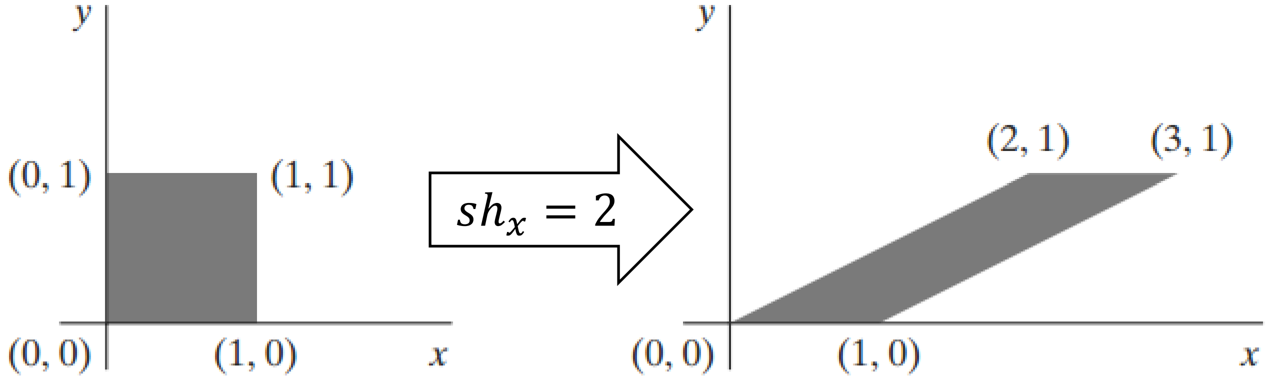


SHEAR

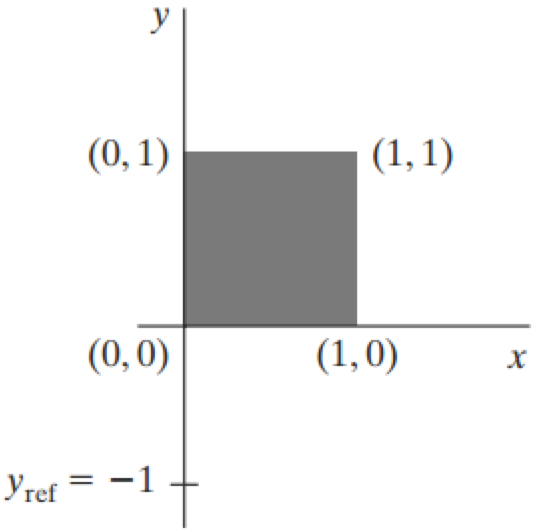
- A transformation that distorts the shape of an object

- An x-direction shear relative to the x axis

$$\begin{bmatrix} 1 & sh_x & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$



SHEAR W.R.TO A REFERENCE

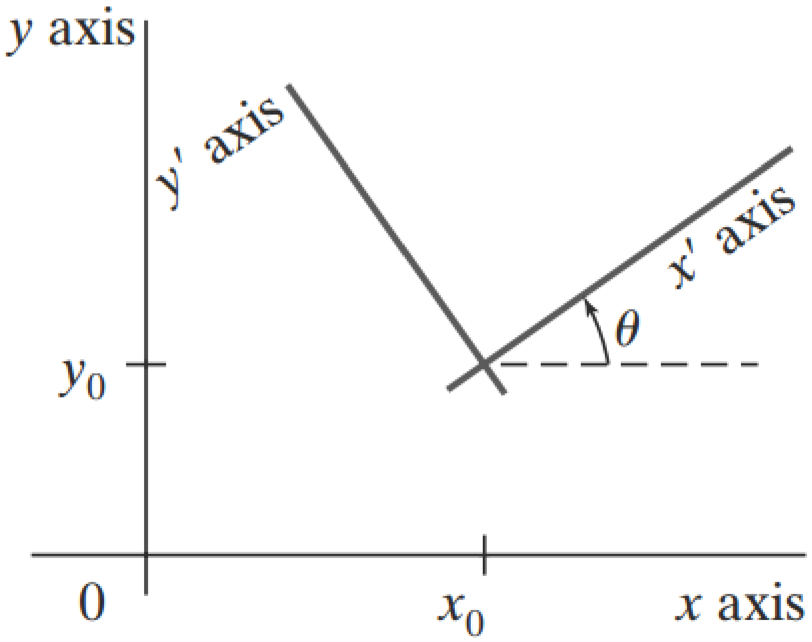


x-direction shears relative to other reference lines

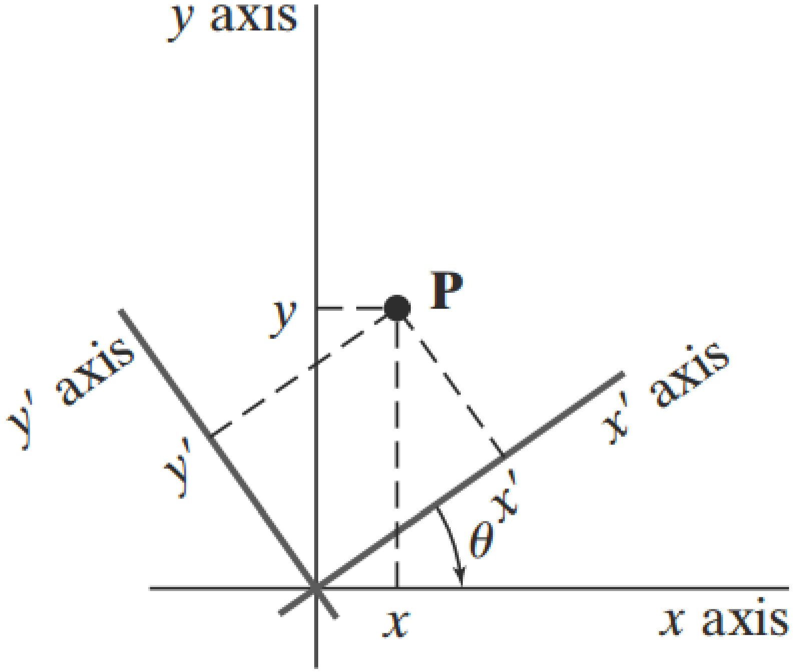
Composite matrix:
$$\begin{bmatrix} 1 & sh_x & -sh_x \cdot y_{\text{ref}} \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

2D COORDINATE TRANSFORMATION

- Converting from one 2D Cartesian frame to the other

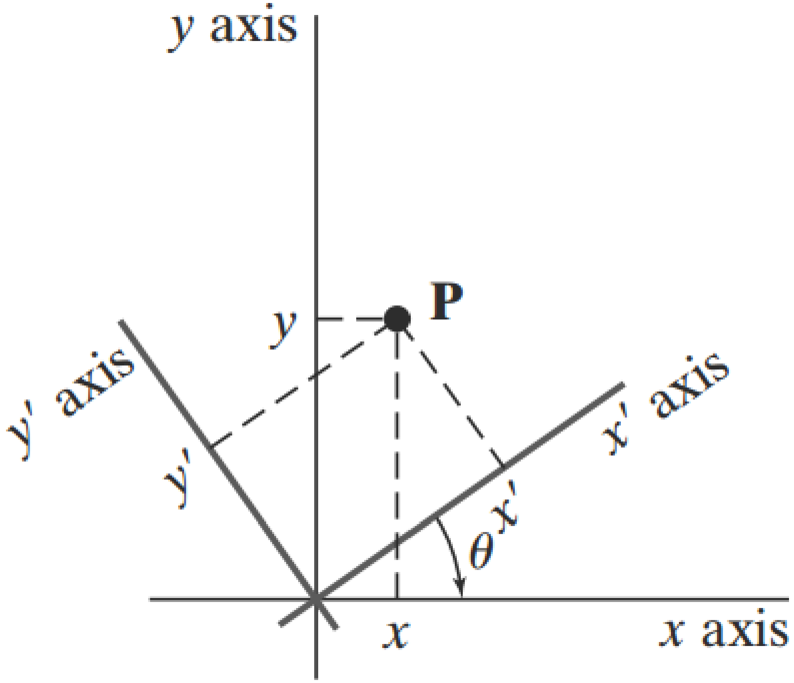


2D COORDINATE TRANSFORMATION (1)



$$\mathbf{T}(-x_0, -y_0) = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

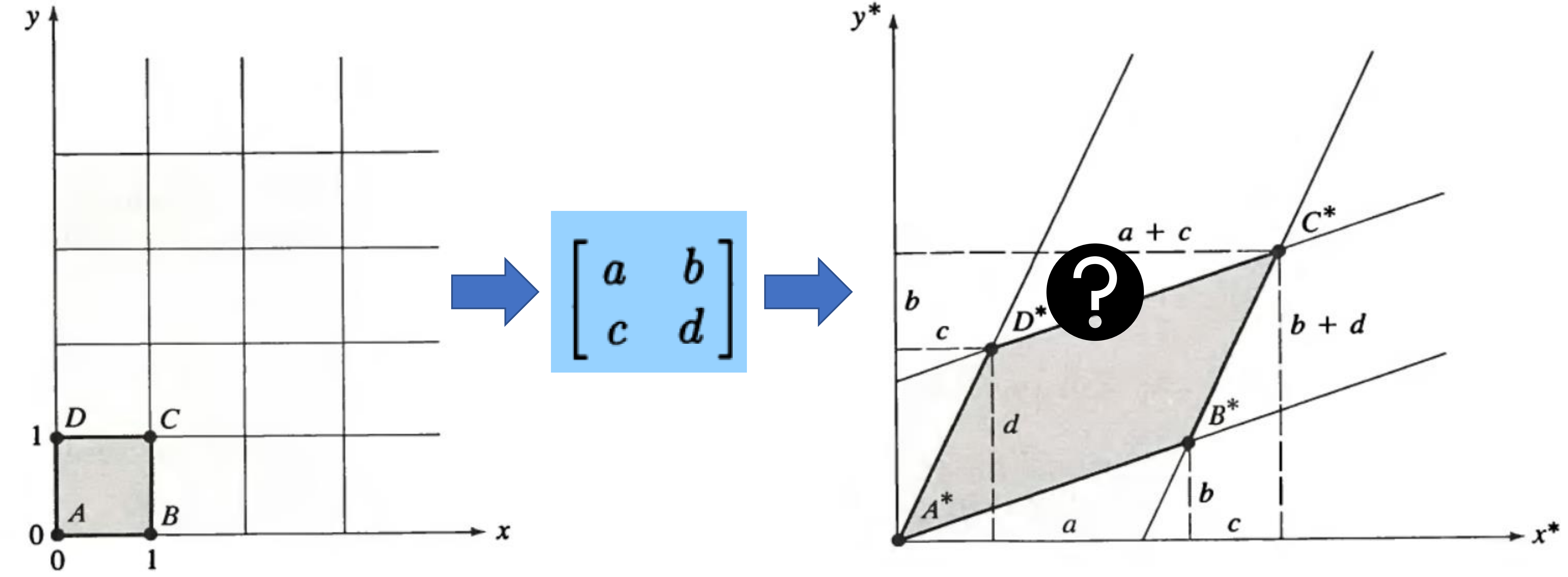
2D COORDINATE TRANSFORMATION (2)



$$\mathbf{T}(-x_0, -y_0) = \begin{bmatrix} 1 & 0 & -x_0 \\ 0 & 1 & -y_0 \\ 0 & 0 & 1 \end{bmatrix}$$

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

TRANSFORMATION OF THE UNIT SQUARE



$$A_p = (a+c)(b+d) - \frac{1}{2}(ab) - \frac{1}{2}(cd) - \frac{c}{2}(b+b+d) - \frac{b}{2}(c+a+c)$$

$$\boxed{A_p = ad - bc = \det \begin{bmatrix} a & b \\ c & d \end{bmatrix}}$$

FLAME UNIVERSITY

EVERLASTING learning

THANK YOU