GALGOTIAS UNIVERSITY

**School of Computing Science and Engineering**

Program: B.C.A.

Course Code: BCAS3003

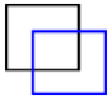
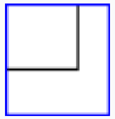
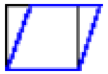
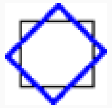
Course Name: Computer Graphics

# Affine Transformation

- Affine transformation is a linear mapping method that preserves points, straight lines, and planes

**Properties**:

- Linearity: Collinearity between points is preserved
- Parallelism: Parallel lines remain parallel
- Lines map to lines

| Affine Transform | Example | Transformation Matrix | |
|---|---|---|---|
| Translation |  | $\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ t_x & t_y & 1 \end{bmatrix}$ | $t_x$ specifies the displacement along the $x$ axis<br>$t_y$ specifies the displacement along the $y$ axis. |
| Scale |  | $\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $s_x$ specifies the scale factor along the $x$ axis<br>$s_y$ specifies the scale factor along the $y$ axis. |
| Shear |  | $\begin{bmatrix} 1 & sh_y & 0 \\ sh_x & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $sh_x$ specifies the shear factor along the $x$ axis<br>$sh_y$ specifies the shear factor along the $y$ axis. |
| Rotation |  | $\begin{bmatrix} \cos(q) & \sin(q) & 0 \\ -\sin(q) & \cos(q) & 0 \\ 0 & 0 & 1 \end{bmatrix}$ | $q$ specifies the angle of rotation. |

# Affine Transformation

- Consider a point P(x,y) then affine transformation of P are all transformation that can be written as
- P' = $\begin{bmatrix} ax + by + c \\ dx + ey + f \end{bmatrix}$
- Where a, b, c, d, e, and f are scalar

1. **Translation**: if a,e=1 and b, d=0
2. **Scaling**: if b, d, c ,f =0
3. **Rotation**: if a,e =Cosɵ and b=-Sinɵ, d-Sinɵ and c,f=0
4. **Shearing**: if a,e=1 and c,f=0

# Transformation functions

**Introduction:** Graphics packages can be structured so that separate commands are provided to a user for each of the basic transformation operations, as in procedure transform object. A composite transformation is then set up by referencing individual functions in the order required for the transformation sequence. An alternate formulation is to provide users with a single transformation function that includes parameters for each of the basic transformations. The output of this function is the composite transformation matrix for the specified parameter values. Both options are useful. Separate functions are convenient for simple transformation operations, and a composite function can provide an expedient method for specifying complex transformation sequences.

The PHIGS library provides users with both options. Individual commands for generating the basic transformation matrices are

**translate (translatevector, matrixTranslate)**

**rotate (theta, matrixRotate)**

**scale (scalevector, matrixscale)**

**composeMatrix (matrix2, matrixl, matrixout)**

Where elements of the composite output matrix are calculated by post multiplying matrix2 by matrix1. A composite transformationmatrix to perform a combination scaling, rotation, and translation is produced with the function

# Cont..

- <u>Equations:-</u>

$$P = (x, y)$$
$$R = (\theta)$$
$$x = r\cos\phi$$
$$y = r\sin\phi$$
$$x' = r\cos(\phi + \Theta)$$
$$y' = r\cos(\phi + \Theta)$$

$$x' = r\cos\phi\cos\theta - r\sin\phi\sin\theta$$
$$y' = r\cos\phi\sin\theta + r\sin\phi\cos\theta$$
$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$
$$P' = R \cdot P$$

# Cont..

- The **anti-clockwise** rotation matrix from above becomes:

$$R(\theta)=\begin{bmatrix} x' \\ y' \end{bmatrix}=\begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}\begin{bmatrix} x \\ y \end{bmatrix}\rightarrow\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}=\begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$P = R(\theta)\bullet P \rightarrow P = R(\theta)\bullet P$$

# Scaling

- A scaling can be represented by a scaling matrix. To scale an object by a vector v = ($v_x$, $v_y$, $v_z$), each point p = ($p_x$, $p_y$, $p_z$) would need to be multiplied with this scaling matrix.

$$S_v = \begin{bmatrix} v_x & 0 & 0 \\ 0 & v_y & 0 \\ 0 & 0 & v_z \end{bmatrix}.$$

# Cont..

- As shown below, the multiplication will give the expected result:

$$S_v p = \begin{bmatrix} v_x & 0 & 0 \\ 0 & v_y & 0 \\ 0 & 0 & v_z \end{bmatrix} \begin{bmatrix} p_x \\ p_y \\ p_z \end{bmatrix} = \begin{bmatrix} v_x p_x \\ v_y p_y \\ v_z p_z \end{bmatrix}.$$

# Reflection

- To reflect a vector about a line that goes through the origin, let be a vector in the direction of the line:
- To reflect a point through a plane *ax* + *by* + *cz* = 0 (which goes through the origin), one can use , where is the 3x3 identity matrix and is the three-dimensional unit vector for the surface normal of the plane.
- To reflect a vector about a line that goes through the origin, let be a vector in the direction of the line:

# Cont..

$$A = \frac{1}{\|\vec{l}\|^2} \begin{bmatrix} l_x^2 - l_y^2 & 2l_x l_y \\ 2l_x l_y & l_y^2 - l_x^2 \end{bmatrix}$$

- To reflect a point through a plane (which goes through the origin), one can use , where is the 3x3 identity matrix and is the three-dimensional unit vector for the surface normal of the plane. If the L2 norm of and is unity, the transformation matrix can be expressed as:

# Cont..

$$A = \begin{bmatrix} 1-2a^2 & -2ab & -2ac \\ -2ab & 1-2b^2 & -2bc \\ -2ac & -2bc & 1-2c^2 \end{bmatrix}$$

- Note that these are particular cases of a Householder reflection in two and three dimensions. A reflection about line or plane that does not go through the origin is not a linear transformation; it is an affine transformation.

# Composition of 2D Transformations

- There are many situations in which the final transformation of a point is a combination of several ( often many ) individual transformations.
- For example, the position of the finger of a robot might be a function of the rotation of the robots hand, arm, and torso, as well as the position of the robot on the railroad train and the position of the train in the world, and the rotation of the planet around the sun.

# Numerical on 2D Transformation

**Question 1:**

- **Compress the square A(0,0), B(0,1), C(1,1) and D(1,0) half its size.**

**Question 2:**

- **A Triangle A(2,2), B(4,2) and C(4,4) is rotated by an angle 90 degree. Find the transformed coordinates**

**Question 3:**

- **A line A(10,5), B(50,20) is rotated about its mid point by an angle 90 degree. Find the coordinates of transferred point.**

# Numerical on 2D Transformation

**Question 4:**

- **What are the new coordinates of the point A(4,-4) after the rotation by 30 degree**
  - **i. about the origin.**
  - **ii. About a point B(2,1)**

**Question 5:**

- **A triangle A(20,0), B(80,0) and C(50,100) being translated 100 units to the right and 20 units up (tx=100, ty=20) find the new coordinates**

**Question 6:**

- **What are the new coordinates of the point P(10,15) after the mirror reflection about**

**i. X-Axis**

**ii. Y-Axis**

# Thank You