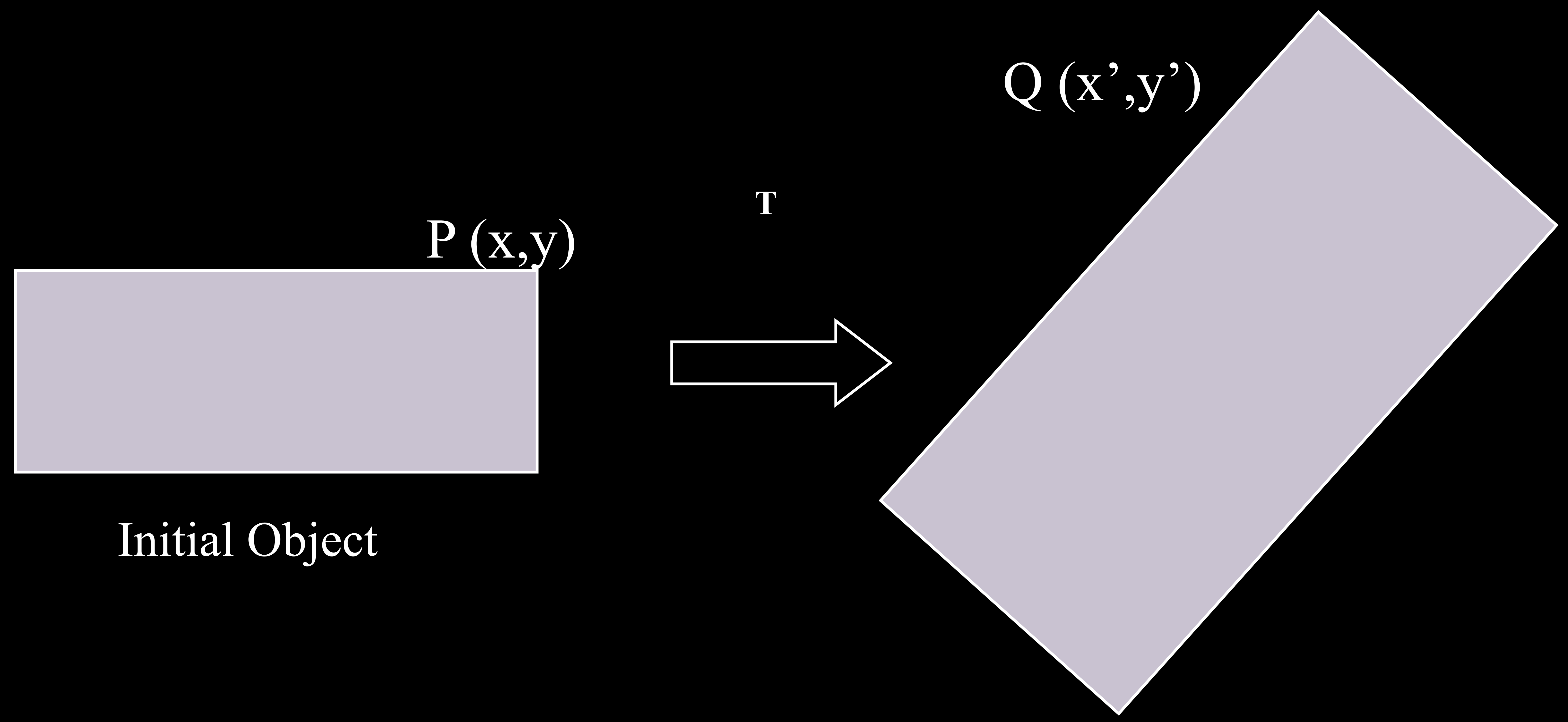
TWO DIMENSIONAL TRANSFORMATION

Transformation

Transform every point on an object according to certain rule.



Initial Object

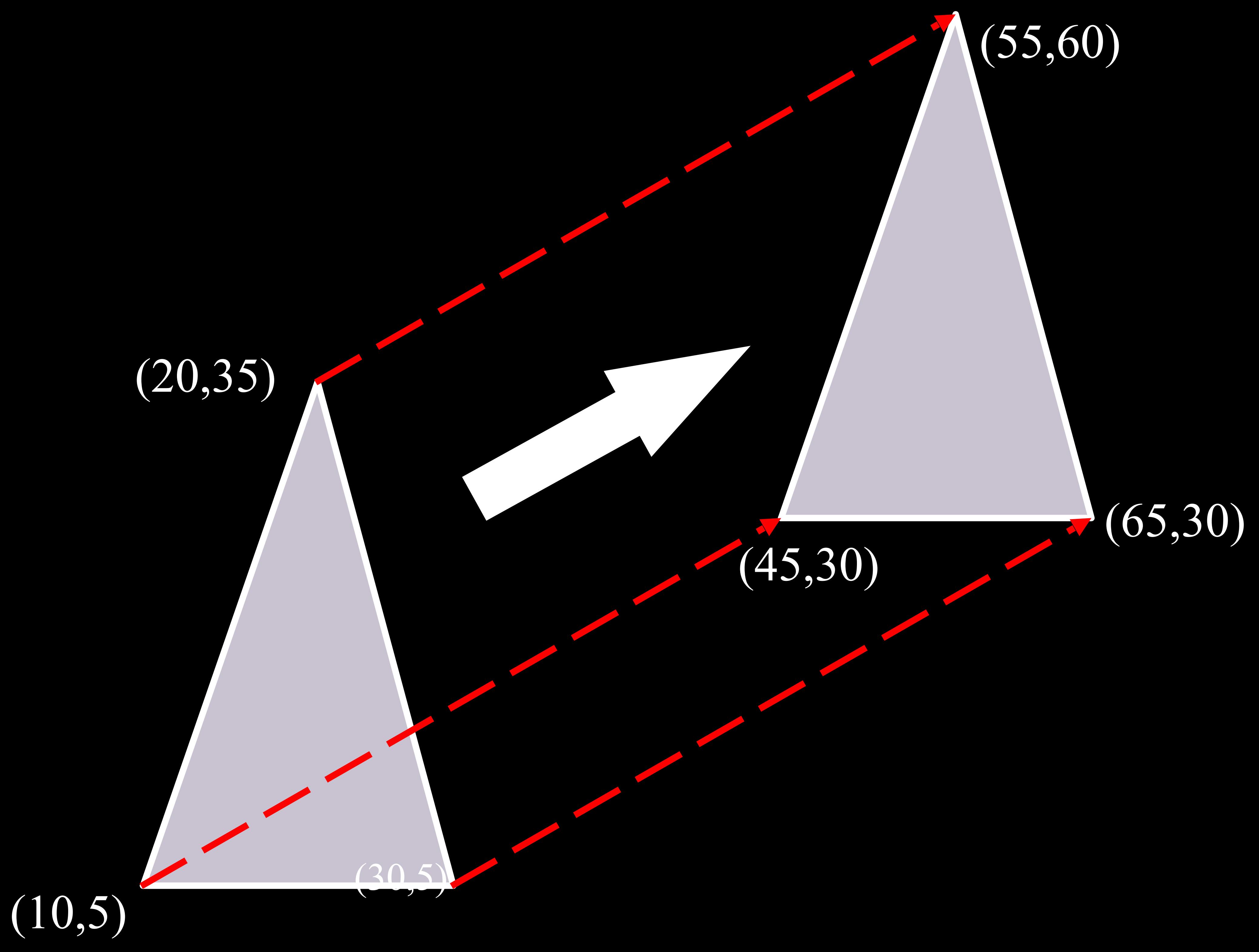
Transformed Object

$x \longrightarrow x'$

$y \longrightarrow y'$

The point Q is the image of P under the transformation T.

Translation



The vector (t_x, t_y) is called the *offset vector*.

$$x' = x + t_x$$

$$y' = y + t_y$$

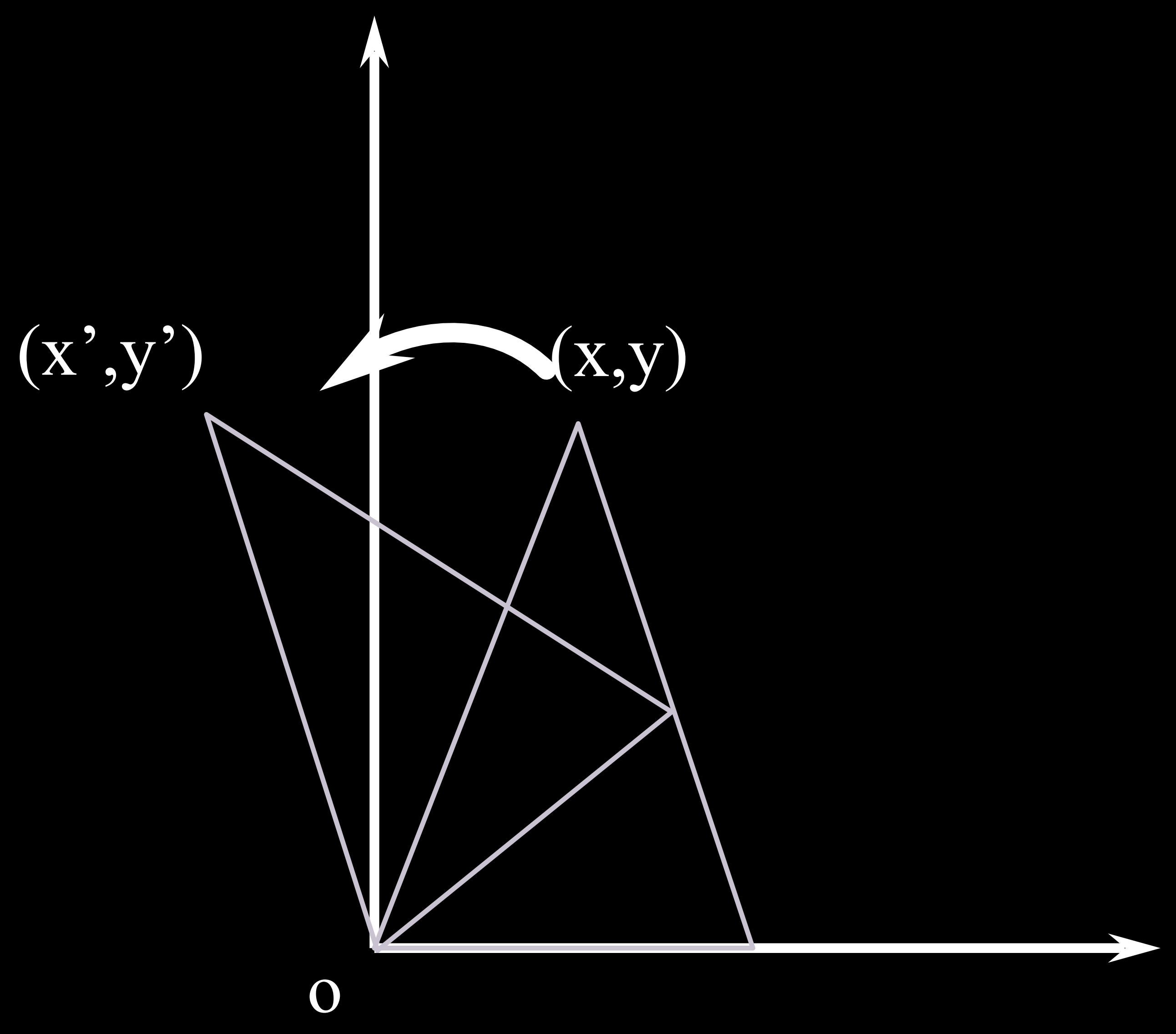
Translation (OpenGL)

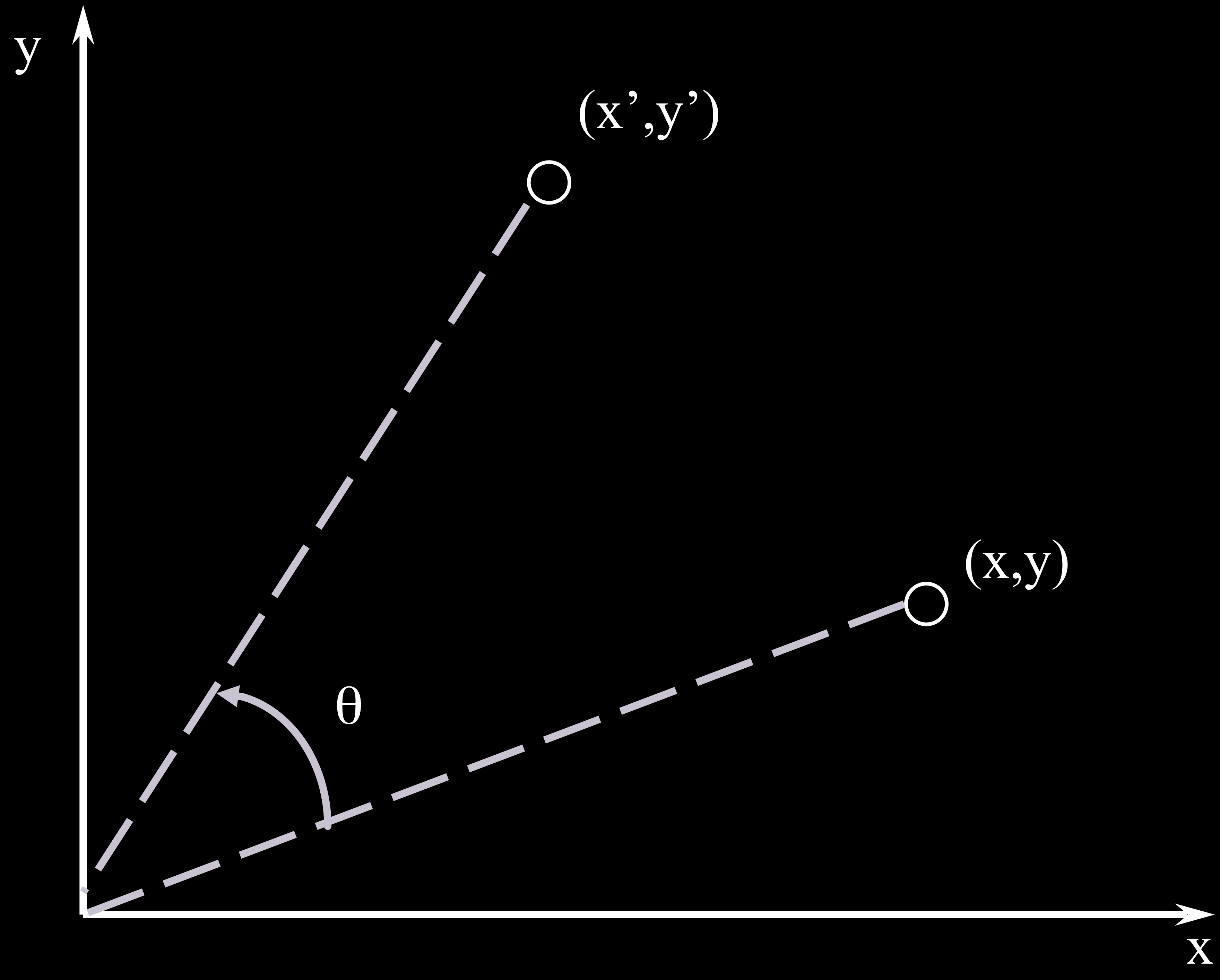
Specifying a 2D-Translation:

```
glTranslatef(tx, ty, 0.0);
```

(The z component is set to 0 for 2D translation).

Rotation About the Origin





$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$

The above 2D rotation is actually a rotation about the z-axis (0,0,1) by an angle θ.

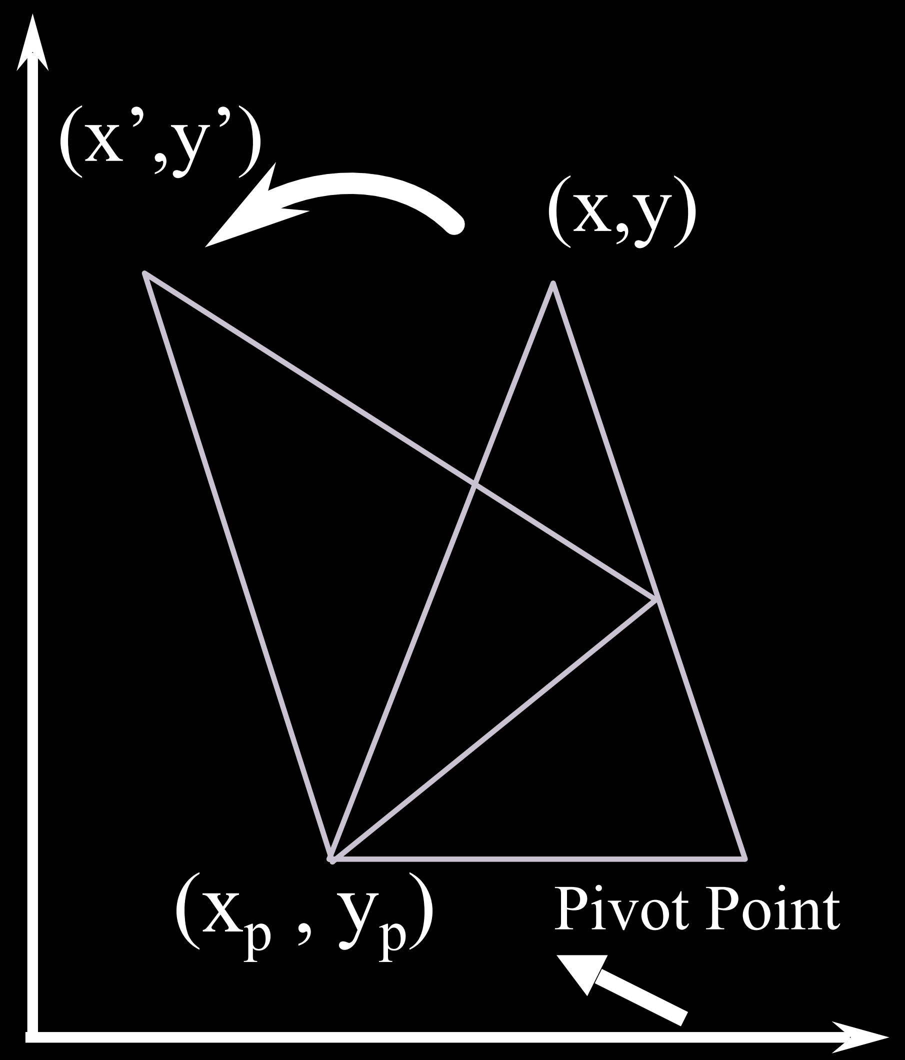
Rotation About the Origin

Specifying a 2D-Rotation about the origin:

```
glRotatef(theta, 0.0, 0.0, 1.0);
```

`theta`: Angle of rotation in degrees.
The above function defines a rotation about the z-axis (0,0,1).

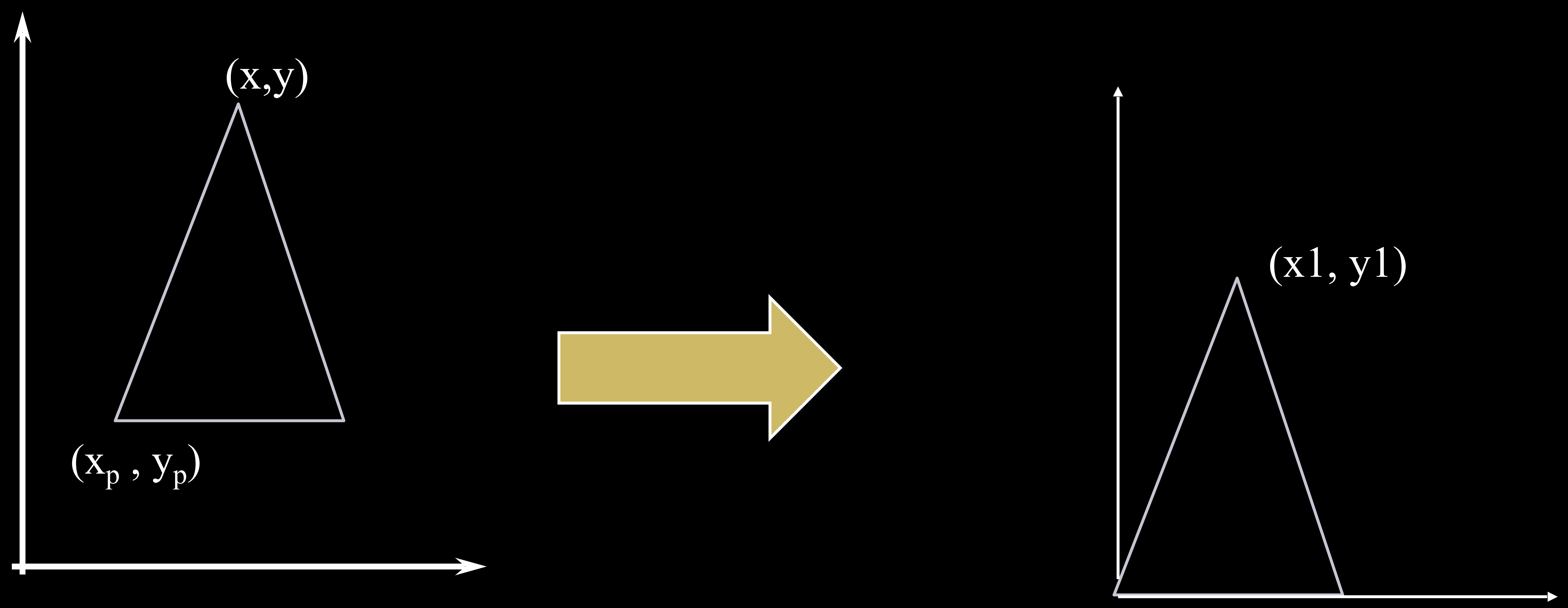
Rotation About a Pivot Point



- Pivot point is the point of rotation
- Pivot point need not necessarily be on the object

Rotation About a Pivot Point

STEP-1: Translate the pivot point to the origin

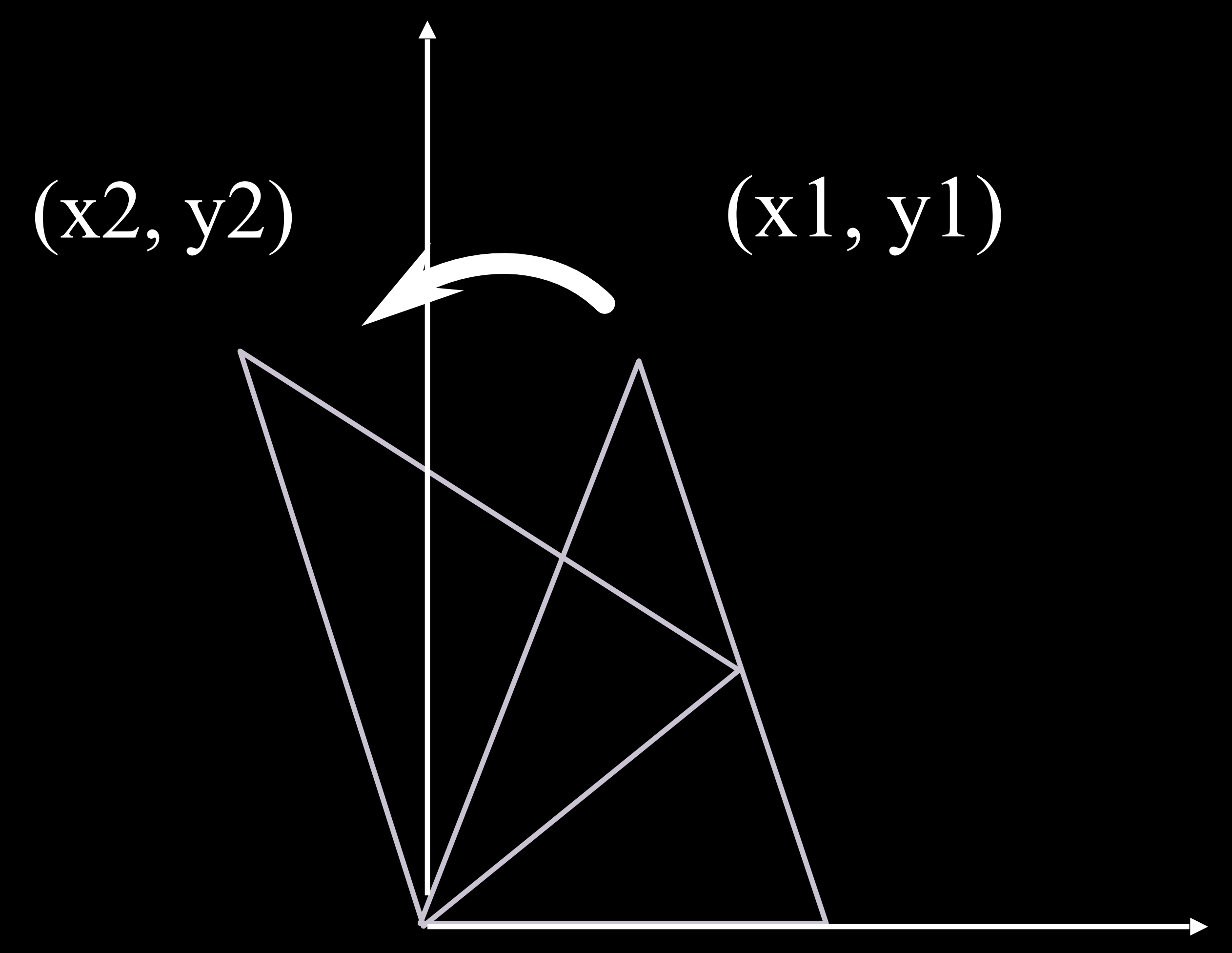


$$x1 = x - x_p$$

$$y1 = y - y_p$$

Rotation About a Pivot Point

STEP-2: **Rotate about the origin**

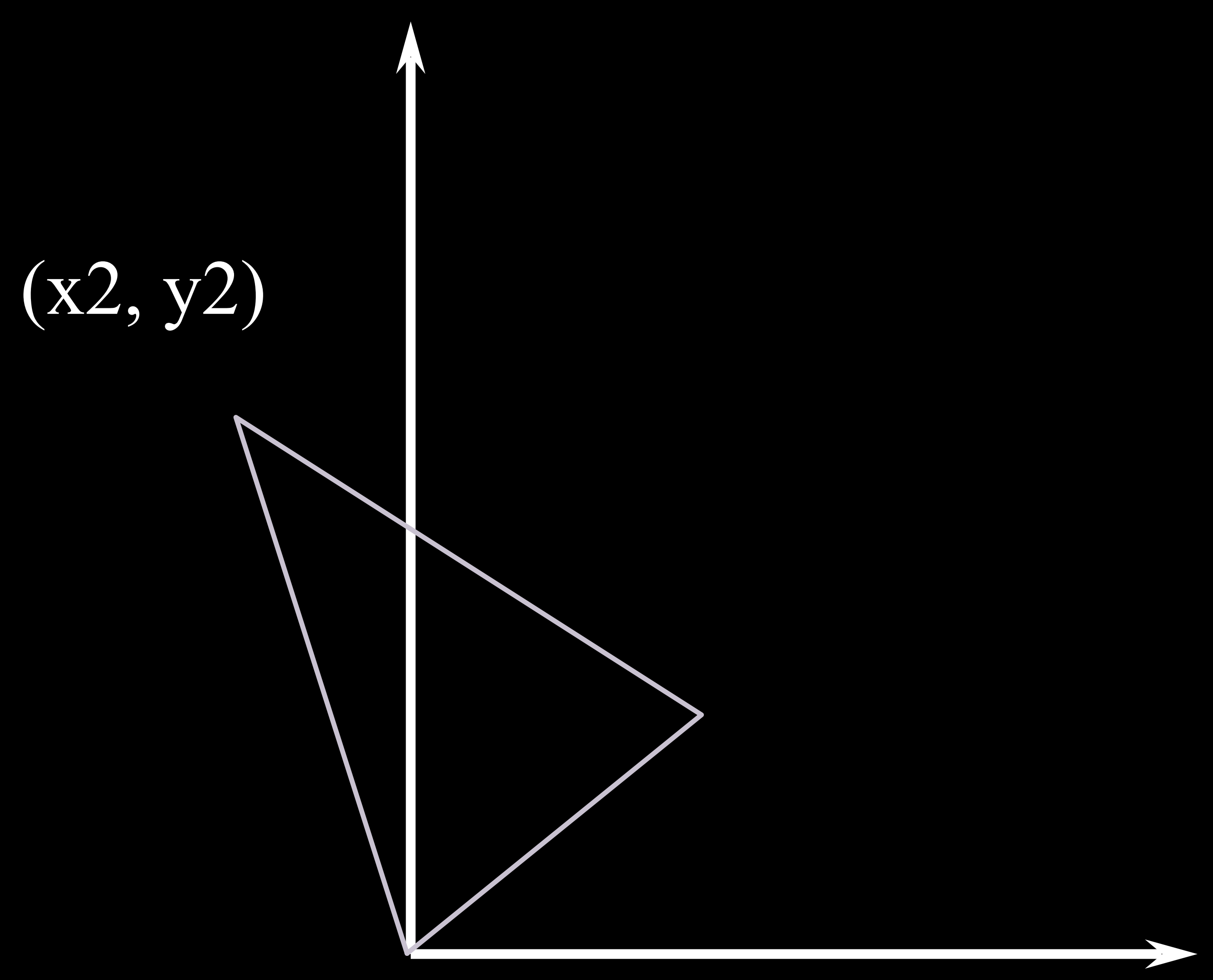


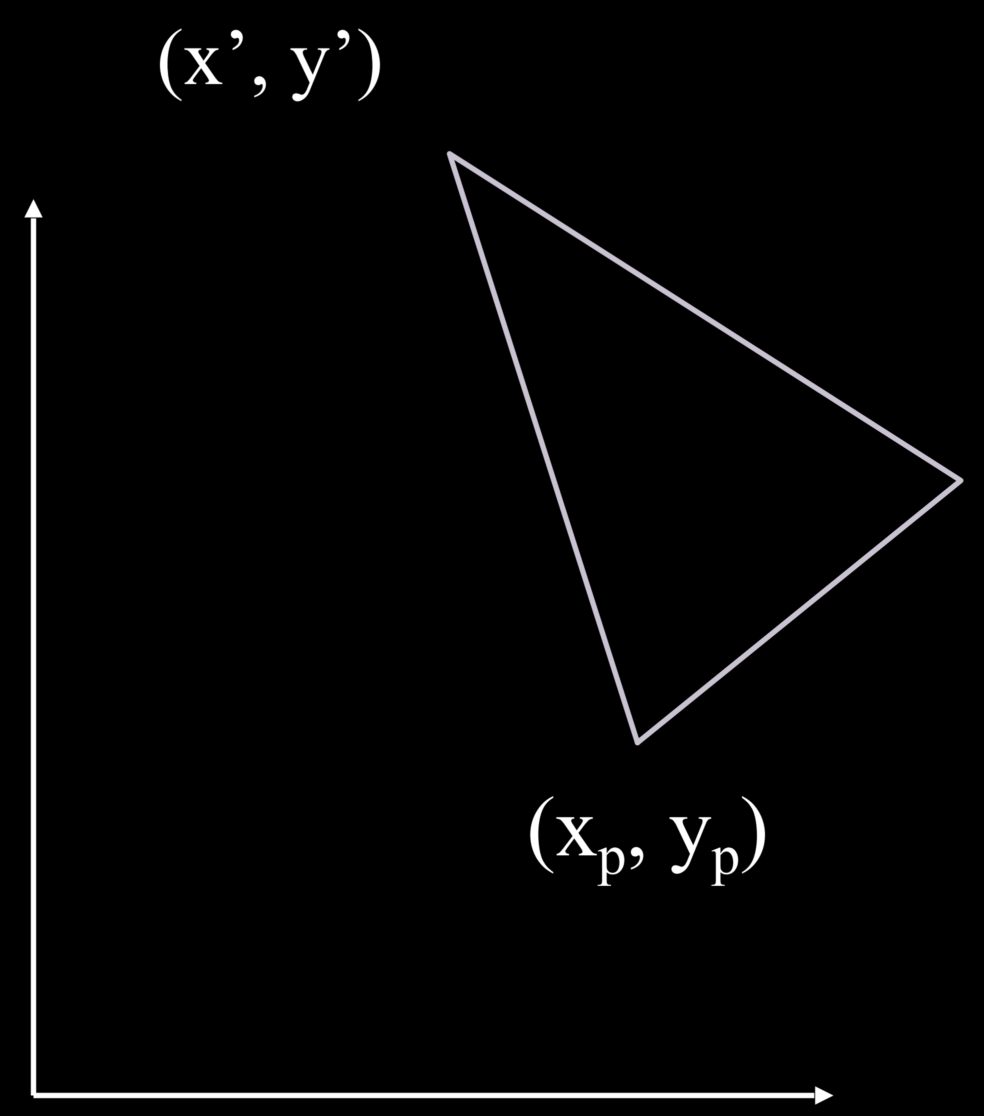
$$x2 = x1\cos\theta - y1\sin\theta$$

$$y2 = x1\sin\theta + y1\cos\theta$$

Rotation About a Pivot Point

<u>STEP-3:</u> **Translate the pivot point to original position**





$$x' = x2 + x_p$$

$$y' = y2 + y_p$$

Rotation About a Pivot Point

$$x' = (x - x_p)\cos\theta - (y - y_p)\sin\theta + x_p$$

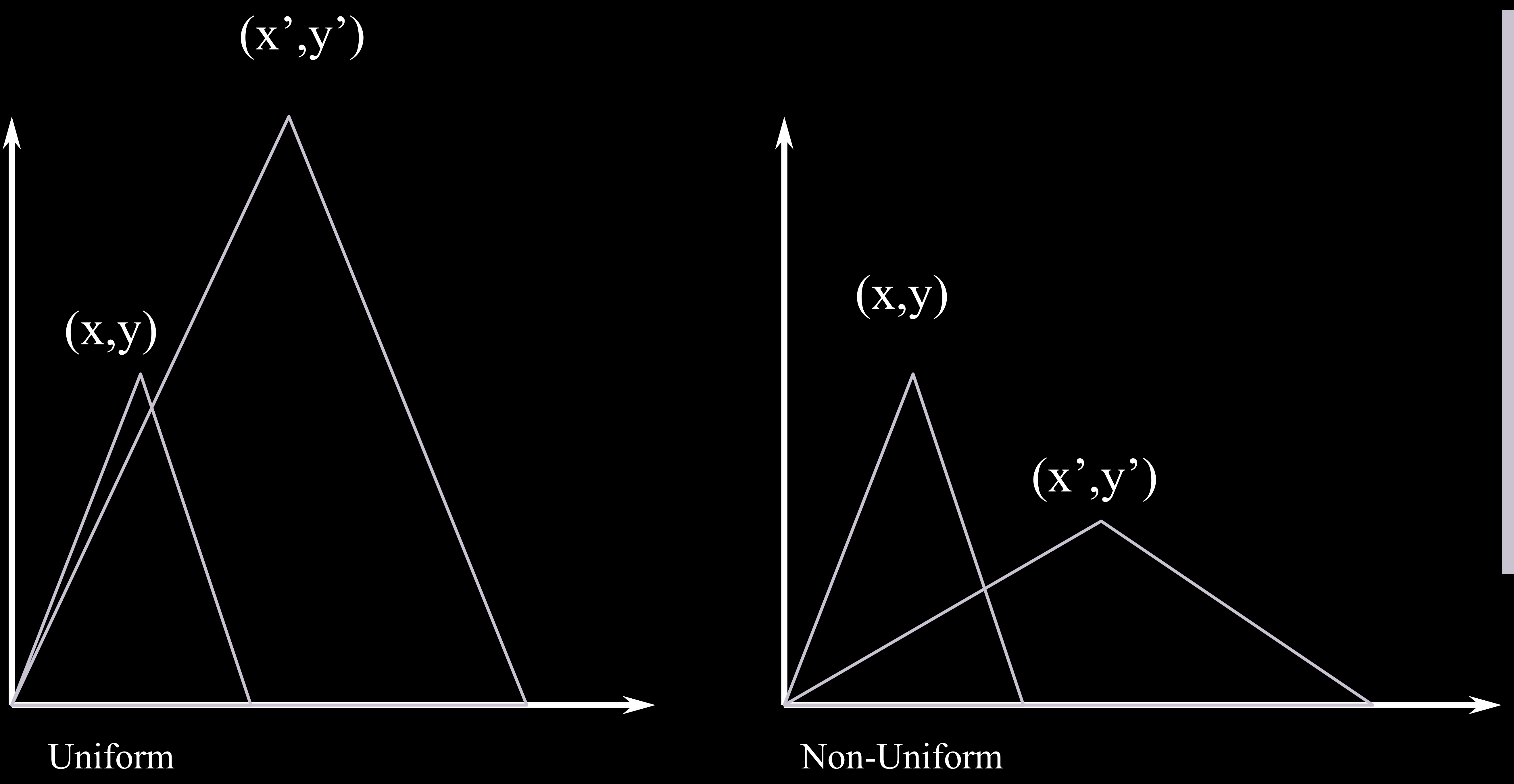
$$y' = (x - x_p)\sin\theta + (y - y_p)\cos\theta + y_p$$

Specifying a 2D-Rotation about a pivot point (xp,yp):

```
glTranslatef(xp, yp, 0);
glRotatef(theta, 0, 0, 1.0);
glTranslatef(-xp, -yp, 0);
```

Note the OpenGL specification of the sequence of transformations in the reverse order !

Scaling About the Origin



$$x' = x.s_x$$

$$y' = y.s_y$$

$$(s_x, s_y > 0)$$

The parameters s_x, s_y are called *scale factors*.

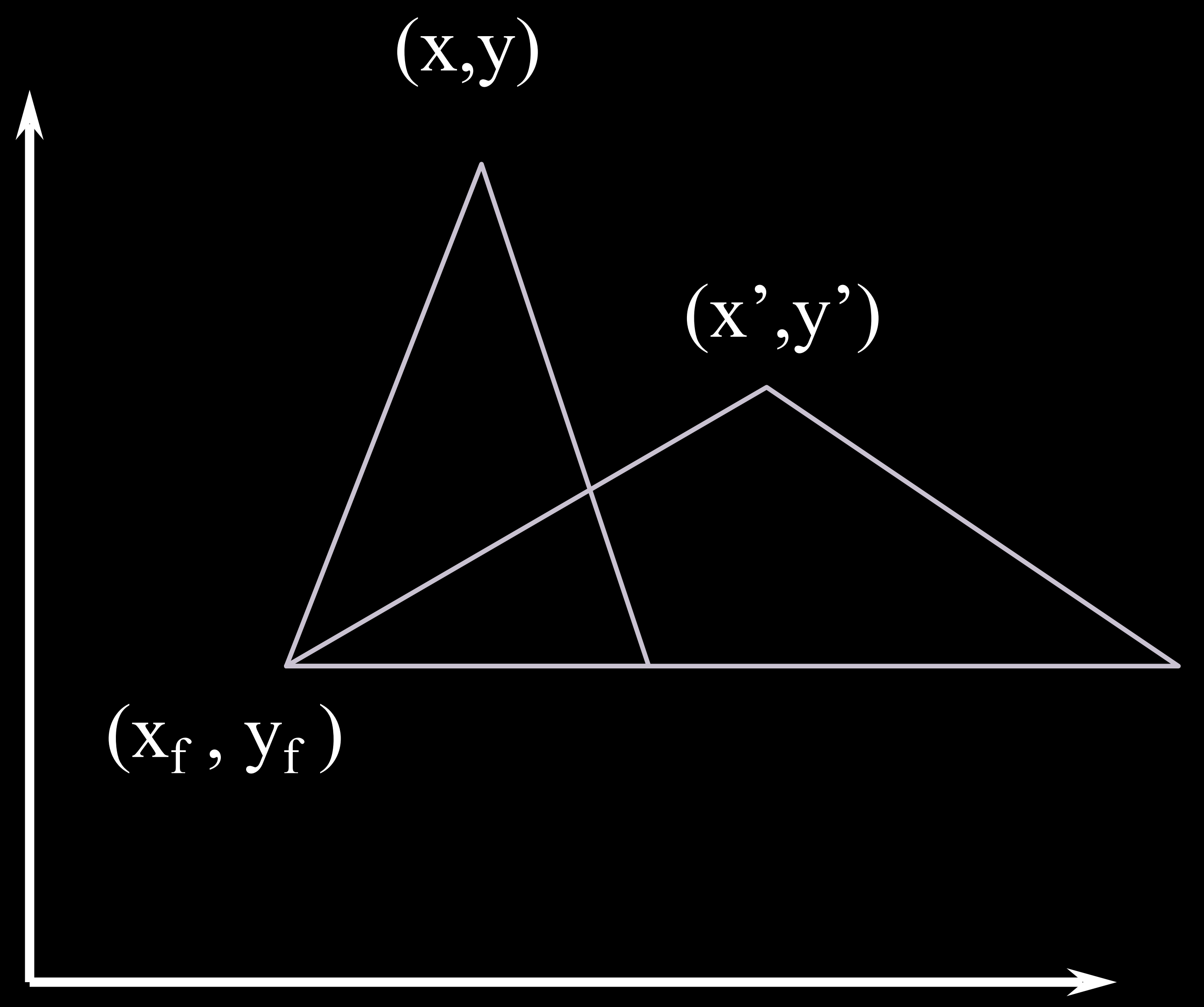
Scaling About the Origin

Specifying a 2D-Scaling with respect to the origin:

```
glScalef(sx, sy, 1.0);
```

`sx, sy`: Scale factors along x, y.
For proper scaling `sx, sy` must be positive.
For 2D scaling, the third scale factor must be set to 1.0.

Scaling About a Fixed Point

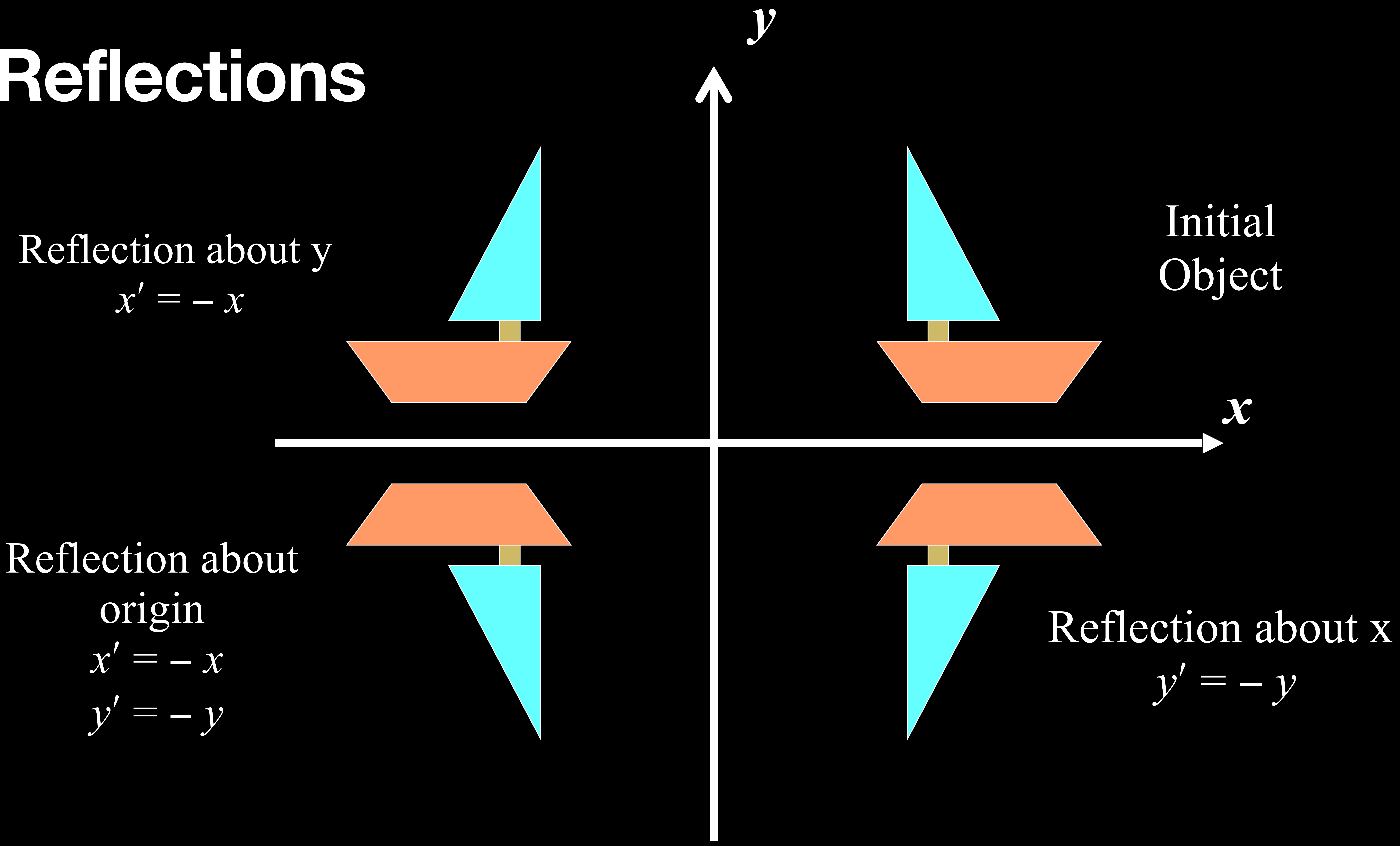


- Translate the fixed point to origin
- Scale with respect to the origin
- Translate the fixed point to its original position.

$$x' = (x - x_f).s_x + x_f$$

$$y' = (y - y_f).s_y + y_f$$

Reflections



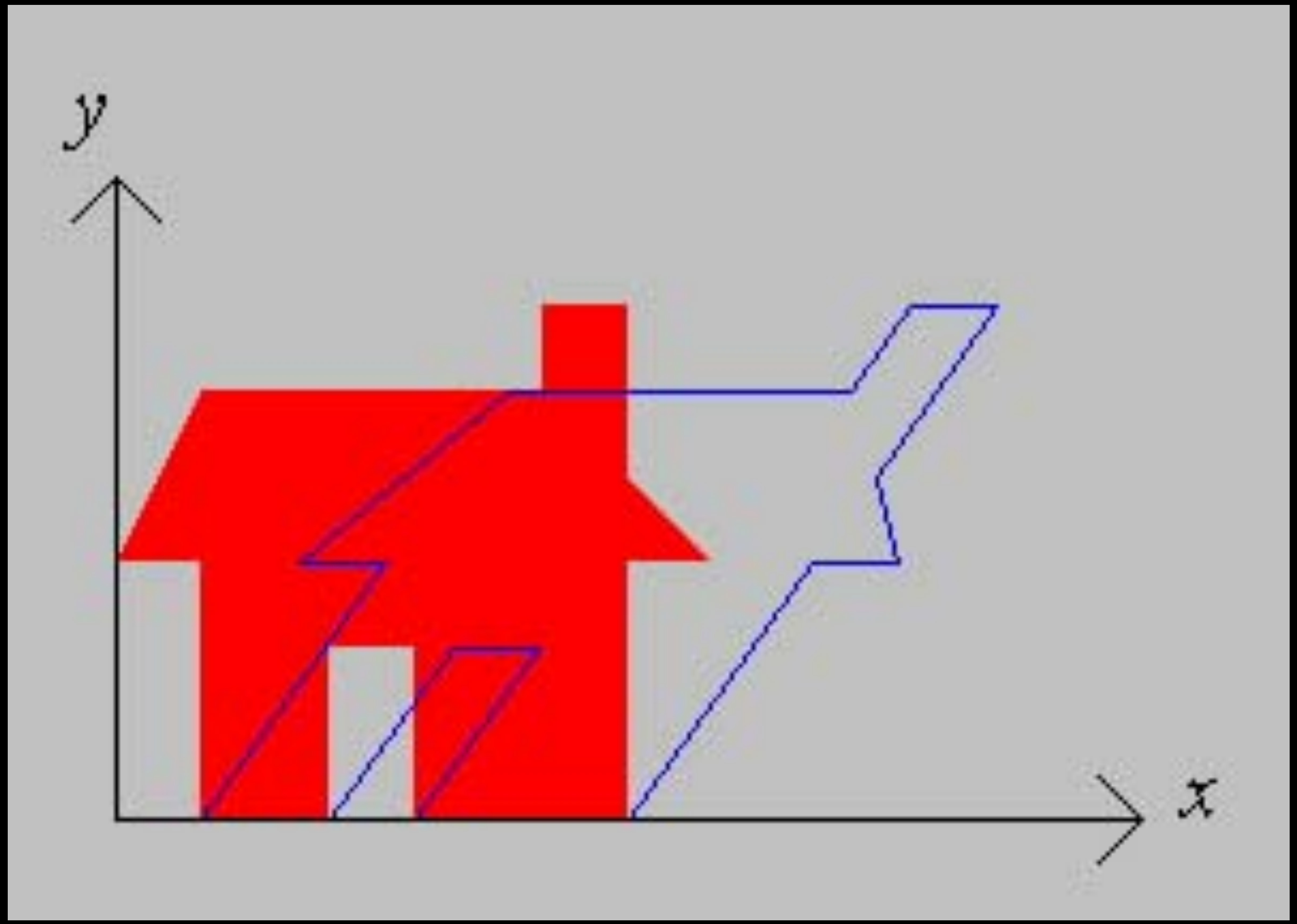
Reflections

Reflection about x: `glScalef(1, -1, 1);`

Reflection about y: `glScalef(-1, 1, 1);`

Reflection about origin: `glScalef(-1, -1, 1);`

Shear



$$x' = x + h_x y$$

$$y' = y$$

- A shear transformation in the x-direction (along x) shifts the points in the x-direction proportional to the y-coordinate.
- The y-coordinate of each point is unaffected.

Matrix Representations

Translation

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

Rotation [Origin]

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

Scaling [Origin]

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

Matrix Representations

Reflection about x

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

Reflection about y

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

Reflection about the Origin

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

Matrix Representations

Shear along x $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & h \\ 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Shear along y $$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ h & 1 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

Example

A A unit square is transformed by a 2×2 transformation matrix. The resulting position vectors are:

$$[x'] = \begin{bmatrix} 0 & 0 \\ 2 & 3 \\ 8 & 4 \\ 6 & 1 \end{bmatrix}$$

Determine the transformation matrix used.

Ans:

$$[x'] = [X][T] = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 1 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} a & b \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ a & b \\ a+c & b+d \\ c & d \end{bmatrix} = \begin{bmatrix} 0 & 0 \\ 2 & 3 \\ 8 & 4 \\ 6 & 1 \end{bmatrix}; \therefore [T] = \begin{bmatrix} 2 & 3 \\ 6 & 1 \end{bmatrix}$$

Example

Show that the shear transformation in x and y directions together is not the same as shear along x followed by shear along y?

Ans:

$$\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix} = \begin{bmatrix} 1+bc & b \\ c & 1 \end{bmatrix} \neq \begin{bmatrix} 1 & b \\ c & 1 \end{bmatrix}$$

$$\begin{bmatrix} 1 & 0 \\ c & 1 \end{bmatrix}\begin{bmatrix} 1 & b \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & b \\ c & 1+bc \end{bmatrix} \neq \begin{bmatrix} 1 & b \\ c & 1 \end{bmatrix}$$

Example

1. A point A has coordinates (2, -3). The point is translated over 3 to the left and up 5 to form A'. A' is reflected across the x-axis to form A''. What are the coordinates of A' and A''?
2. A point P has coordinates (5, -6). The point is reflected across the line $y = -x$ to form P' . P' is rotated about the origin 90∘CW to form P''. What are the coordinates of P' and P''?
3. A point X has coordinates (-1, -8). The point is reflected across the y-axis to form X' . X' is translated over 4 to the right and up 6 to form X''. What are the coordinates of X' and X''?

Example Problem

- **Consider a triangle whose vertices are (2 2), (4 2) and (4 4). Find the concatenated transformation matrix and the transformed vertices for rotatation of 90 about the origin followed by reflection through the line *y* = -*x*. Comment on the sequence of transformations.**

$$[X\}[T_1\}[T_2\} = [X\}[T\} = \begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} -1 & 0 \\ 0 & 1 \end{bmatrix} = \begin{bmatrix} -2 & 2 \\ -4 & 2 \\ -4 & 4 \end{bmatrix}$$

For seeing the effect of changing the sequence of operations let us reverse the order, i.e, first reflection and then rotation.

$$[X\}[T_2\}[T_1\} = [X\}[T\} = \begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} 0 & -1 \\ -1 & 0 \end{bmatrix} \begin{bmatrix} 0 & 1 \\ -1 & 0 \end{bmatrix} = \begin{bmatrix} 2 & 2 \\ 4 & 2 \\ 4 & 4 \end{bmatrix} \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} = \begin{bmatrix} 2 & -2 \\ 4 & -2 \\ 4 & -4 \end{bmatrix}$$

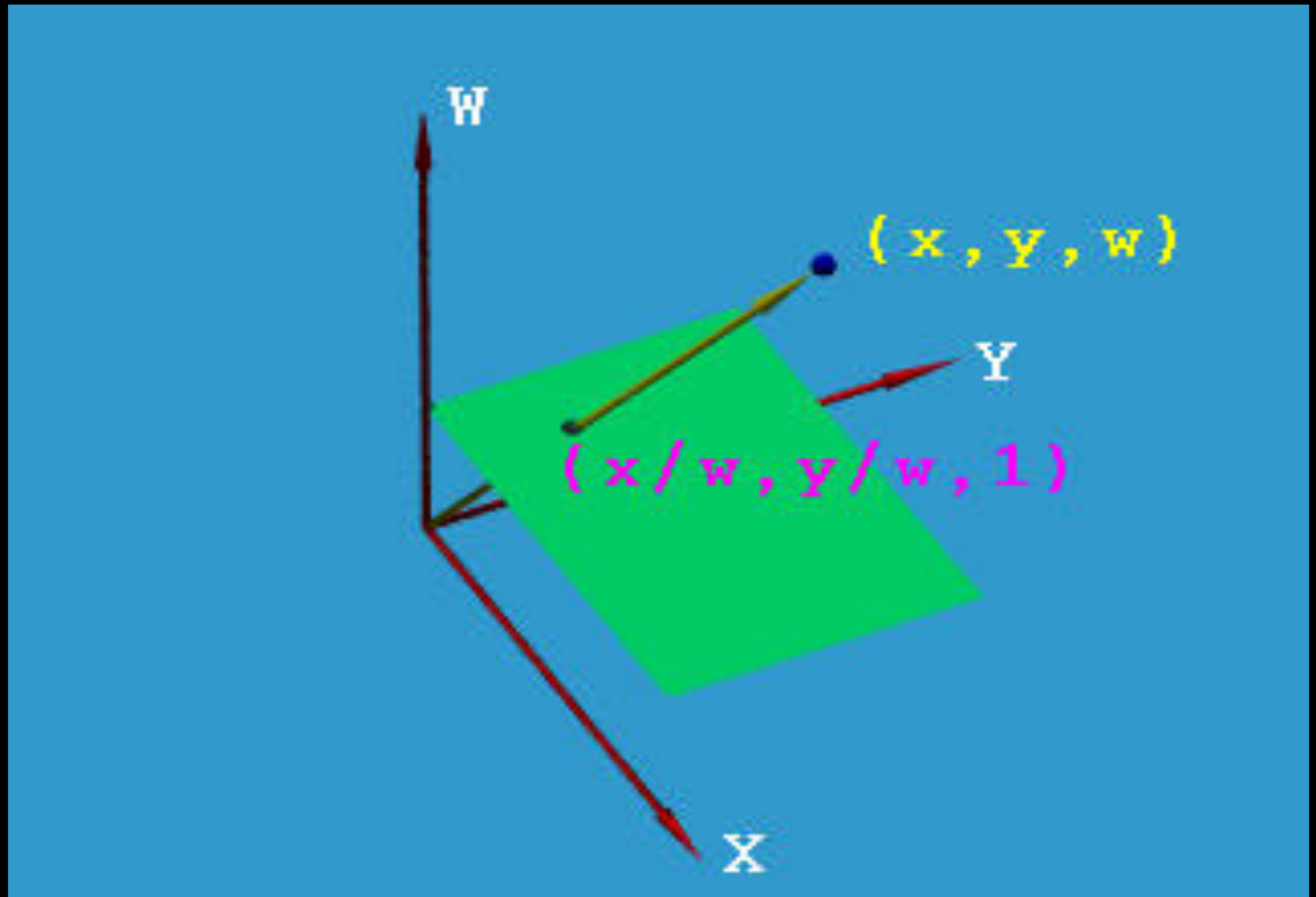
Questions

1. Line JT has coordinates J(−2,−5) and T(2,3). The segment is rotated about the origin 180◦ to form J’ T’ . J’ T’ is translated over 6 to the right and down 3 to form J”T”. What are the coordinates of J’ T’ and J”T”?
2. Line SK has coordinates S(−1,−8) and K(1,2). The segment is translated over 3 to the right and up 3 to form S’ K’ . S’ K’ is rotated about the origin 90◦CCW to form S”K”. What are the coordinates of S’ K’ and S”K”?
3. A point K has coordinates (-1, 4). The point is reflected across the line y = x to form K’ . K’ is rotated about the origin 270◦CW to form K”. What are the coordinates of K’ and K”?
4. Explore what happens when you reflect a shape twice, over a pair of parallel lines. What one transformation could have been performed to achieve the same result?
5. Explore what happens when you reflect a shape twice, over a pair of intersecting lines. What one transformation could have been performed to achieve the same result?
6. Explore what happens when you reflect a shape over the x-axis and then the y-axis. What one transformation could have been performed to achieve the same result?

Homogeneous Coordinates

To obtain square matrices an additional row was added to the matrix and an additional coordinate, the *w*-coordinate, was added to the vector for a point. In this way a point in 2D space is expressed in three-dimensional homogeneous coordinates.

This technique of representing a point in a space whose dimension is one greater than that of the point is called homogeneous representation. It provides a consistent, uniform way of handling affine transformations.



•If we use homogeneous coordinates, the geometric transformations given above can be represented using only a matrix pre-multiplication.

•A composite transformation can then be represented by a product of the corresponding matrices.

Cartesian	Homogeneous

$$(x, y) \longrightarrow (xh, yh, h), h \neq 0$$

$$\left(\frac{a}{c}, \frac{b}{c}\right) \longleftarrow (a, b, c), c \neq 0$$

Examples:	(5, 8)	➡	(15, 24, 3)
	(x, y)		(x, y, 1)

Basic Transformations

Translation
P'=TP

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

Rotation [O]
P'=RP

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

Scaling [O]
P'=SP

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

Inverse of Transformations

If, $\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = [T] \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$ then, $\begin{bmatrix} x \\ y \\ 1 \end{bmatrix} = [T]^{-1} \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$

Examples:

$$T^{-1}(t_x, t_y) = T(-t_x, -t_y)$$

$$R^{-1}(\theta) = R(-\theta)$$

$$S^{-1}(s_x, s_y) = S\left(\frac{1}{s_x}, \frac{1}{s_y}\right)$$

$$H_x^{-1}(h) = H_x(-h)$$

Transformation Matrices

Additional Properties:

$$T(t_x, t_y)T(u_x, u_y) = T(t_x + u_x,\ t_y + u_y)$$

$$R(\theta_1)R(\theta_2) = R(\theta_1 + \theta_2)$$

$$\left|R(\theta)\right| = 1$$

$$S(s_x, s_y)S(w_x, w_y) = S(s_x w_x,\ s_y w_y)$$

Composite Transformations

Transformation T followed by Transformation Q followed by Transformation R:

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = [R][Q][T]\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Example: (Scaling with respect to a fixed point)

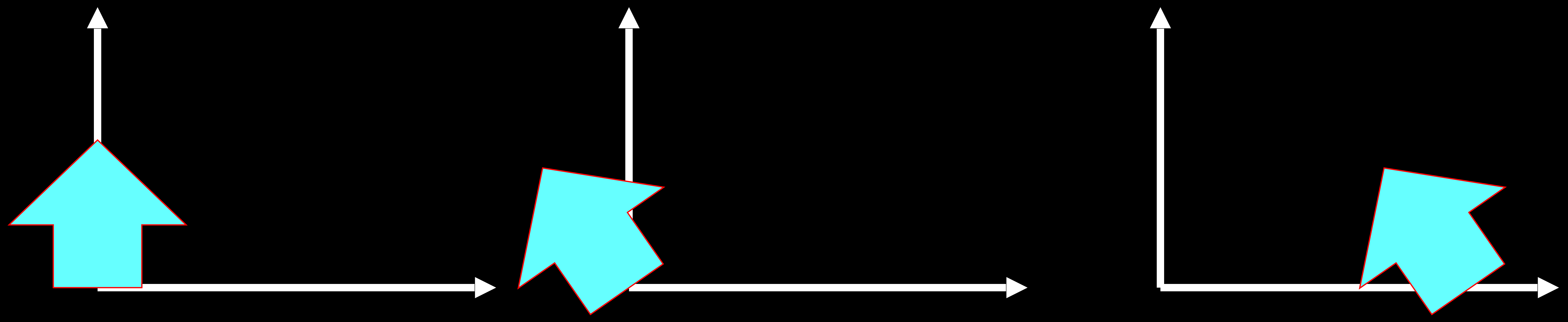
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix}\begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Order of Transformations

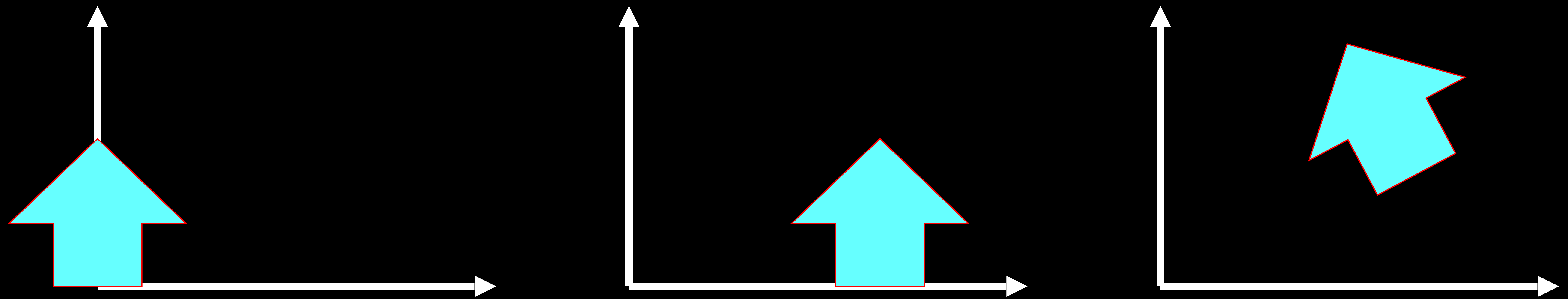
Order of Transformations

In composite transformations, the order of transformations is very important.

Rotation followed by Translation:



Translation followed by Rotation:



Order of Transformations (OpenGL)

OpenGL ***postmultiplies*** the current matrix with the new transformation matrix

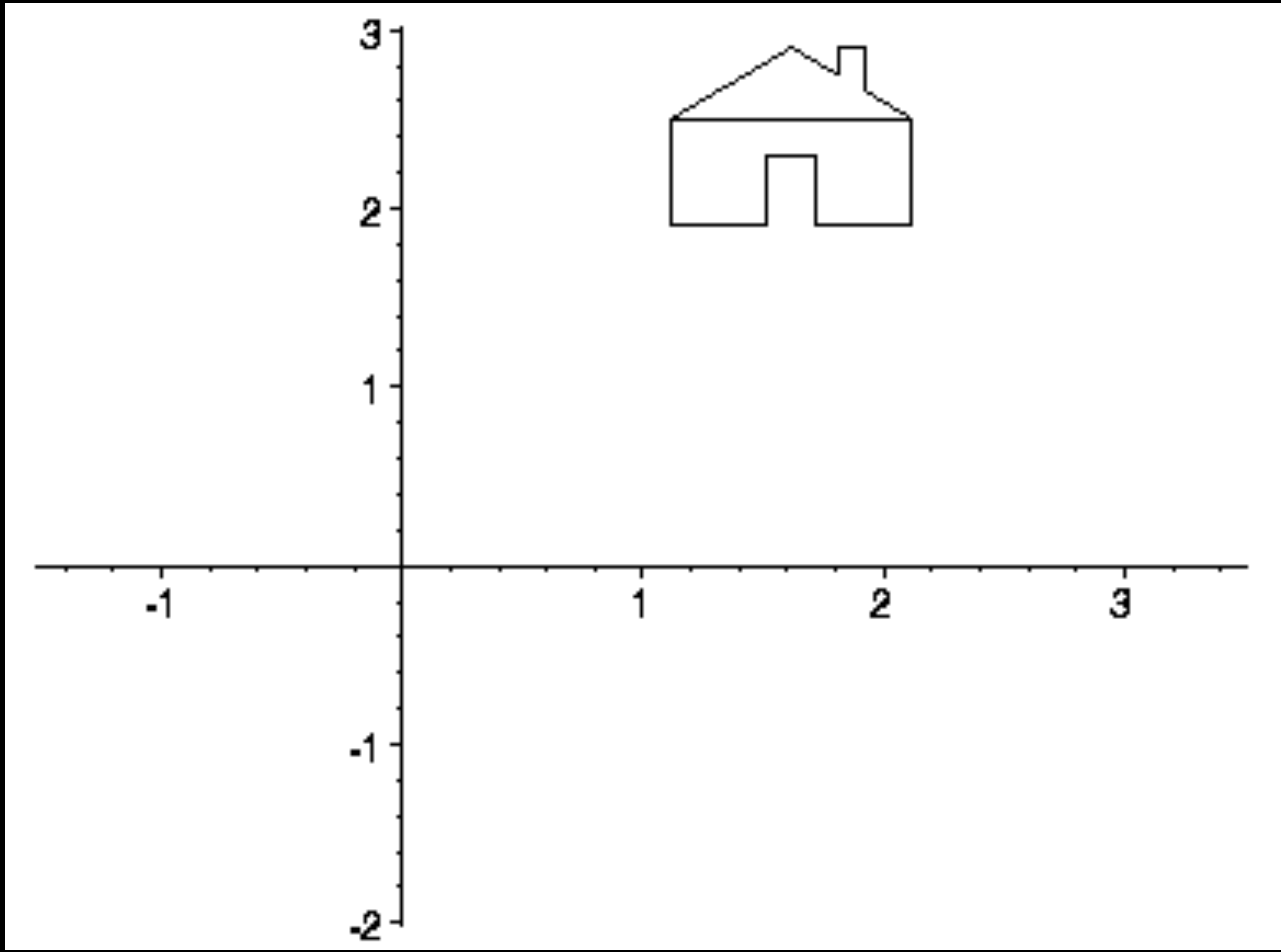
	Current Matrix
`glMatrixMode(GL_MODELVIEW);`	
`glLoadIdentity();`	**[I]**
`glTranslatef(tx, ty, 0);`	**[T]**
`glRotatef(theta, 0, 0, 1.0);`	**[T] [R]**
`glVertex2f(x,y);`	**[T] [R] P**

Rotation followed by Translation !!

3D TRANSFORMATION

TRANSFORMATION

Transformations are a fundamental part of the computer graphics. Transformations are the movement of the object in Cartesian plane .



WHY WE USE TRANSFORMATION

- **Transformation are used to position objects , to shape object , to change viewing positions , and even how something is viewed.**

- **In simple words transformation is used for**

  1) Modeling

  2) Viewing

3D TRANSFORMATION

- When the transformation takes place on a 3D plane .it is called 3D transformation.
- Generalize from 2D by including **z** coordinate Straight forward for translation and scale, rotation more difficult

Homogeneous coordinates: 4 components

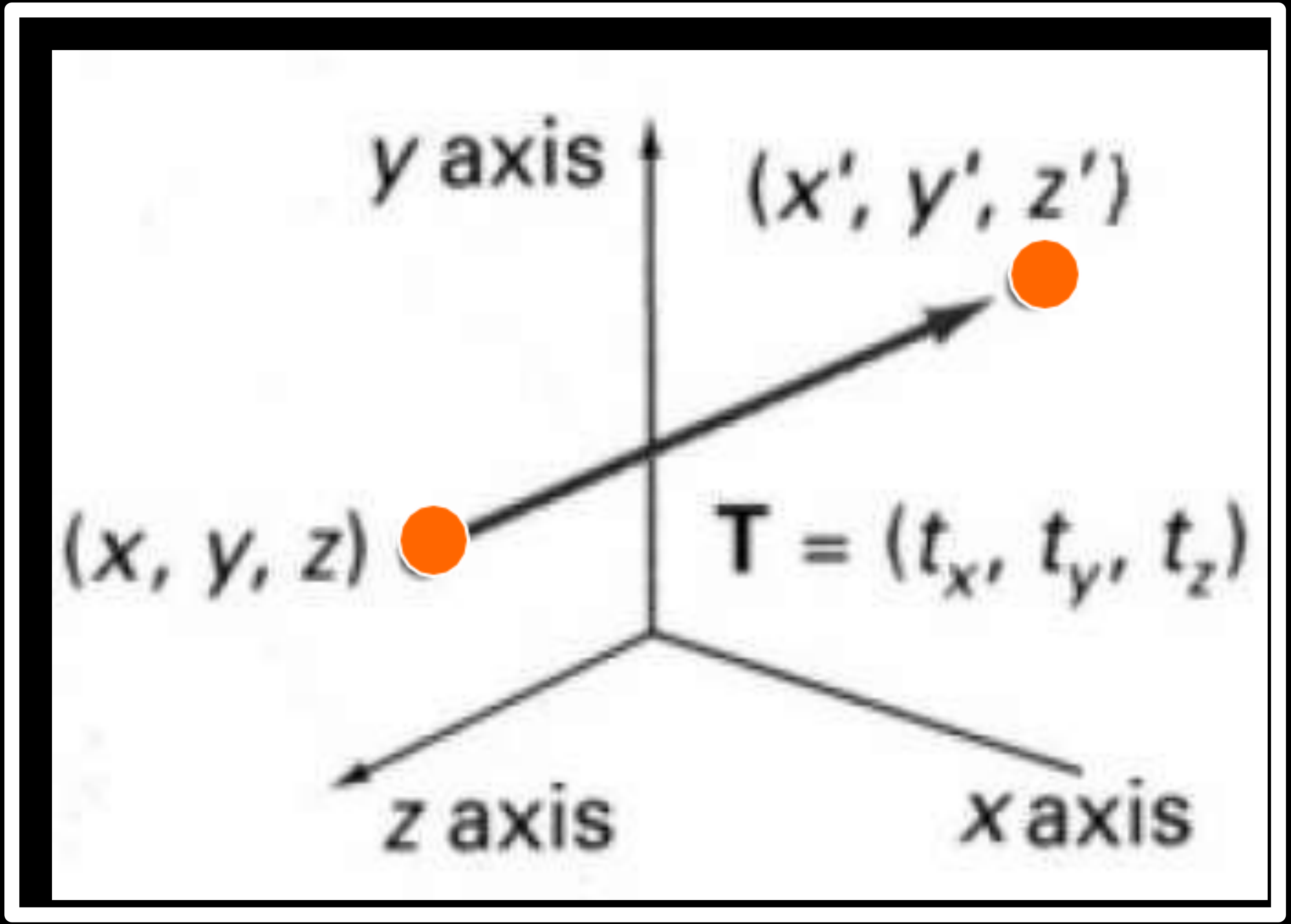
Transformation matrices: 4×4 elements

$$\begin{bmatrix} a & b & c & t_x \\ d & e & f & t_y \\ g & h & i & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3D TRANSLATION

- Moving of object is called translation.
- In 3 dimensional homogeneous coordinate representation , a point is transformed from position P = (x, y , z) to P'=(x', y',z')
- This can be written as:- Using **P' = T .P**

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

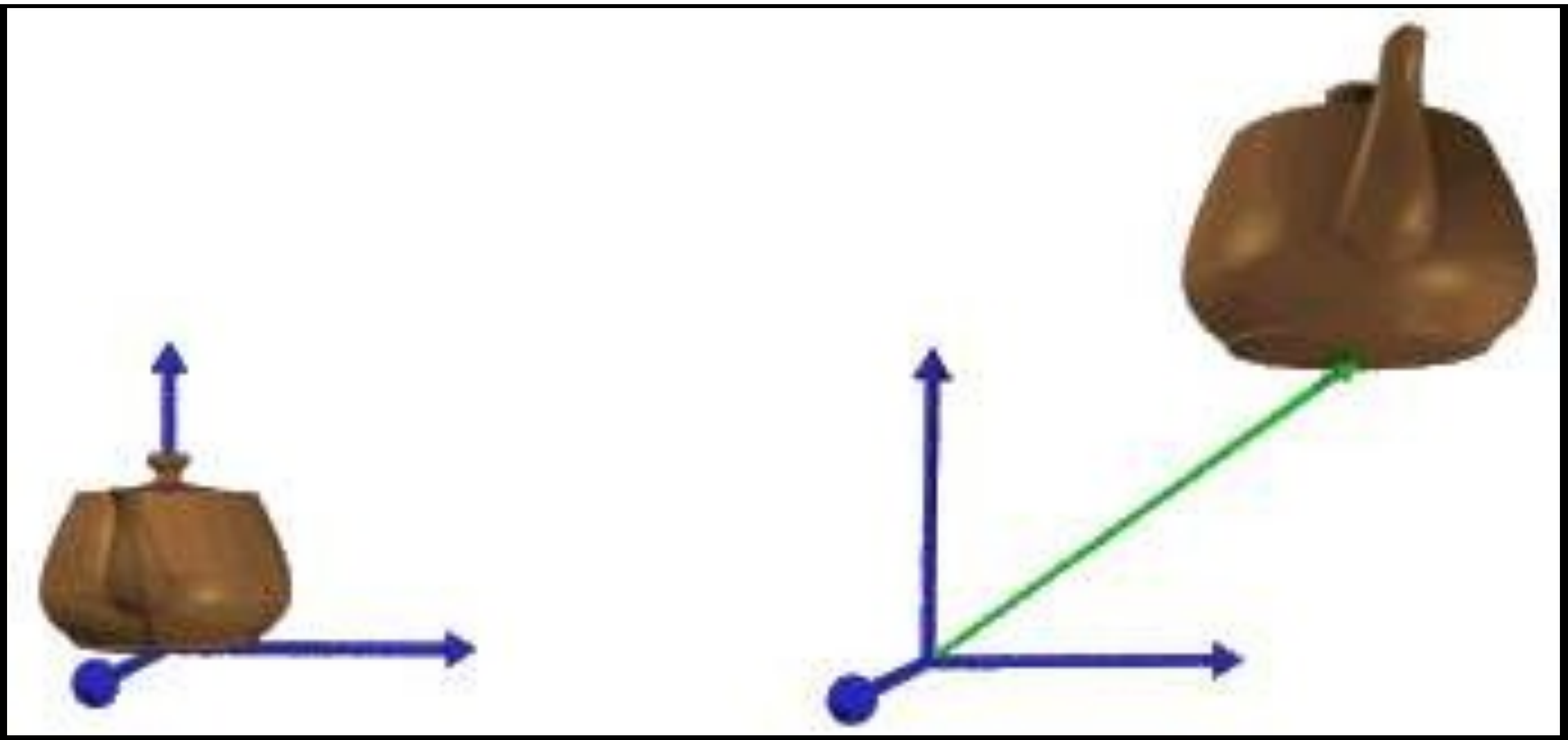


3D TRANSLATION

- The matrix representation is equivalent to the three equation.

$$x' = x + t_x\ ,\ y' = y + t_y\ ,\ z' = z + t_z$$

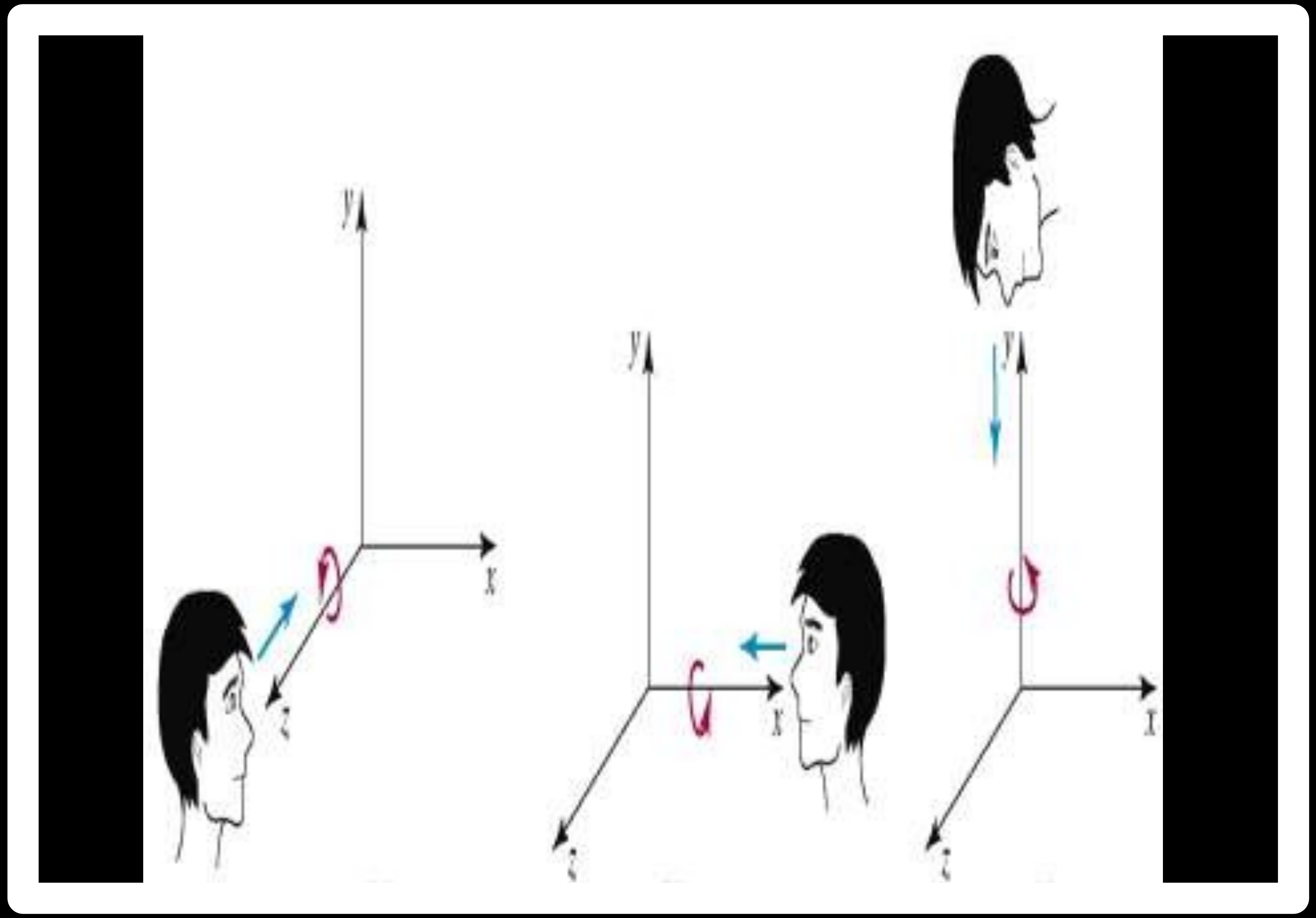
- Where parameter t_x , t_y , t_z are specifying translation distance for the coordinate direction x , y , z are assigned any real value.



3D ROTATION

- Where an object is to be rotated about an axis that is parallel to one of the coordinate axis, we can obtain the desired rotation with the following transformation sequence.
- Coordinate axis rotation :

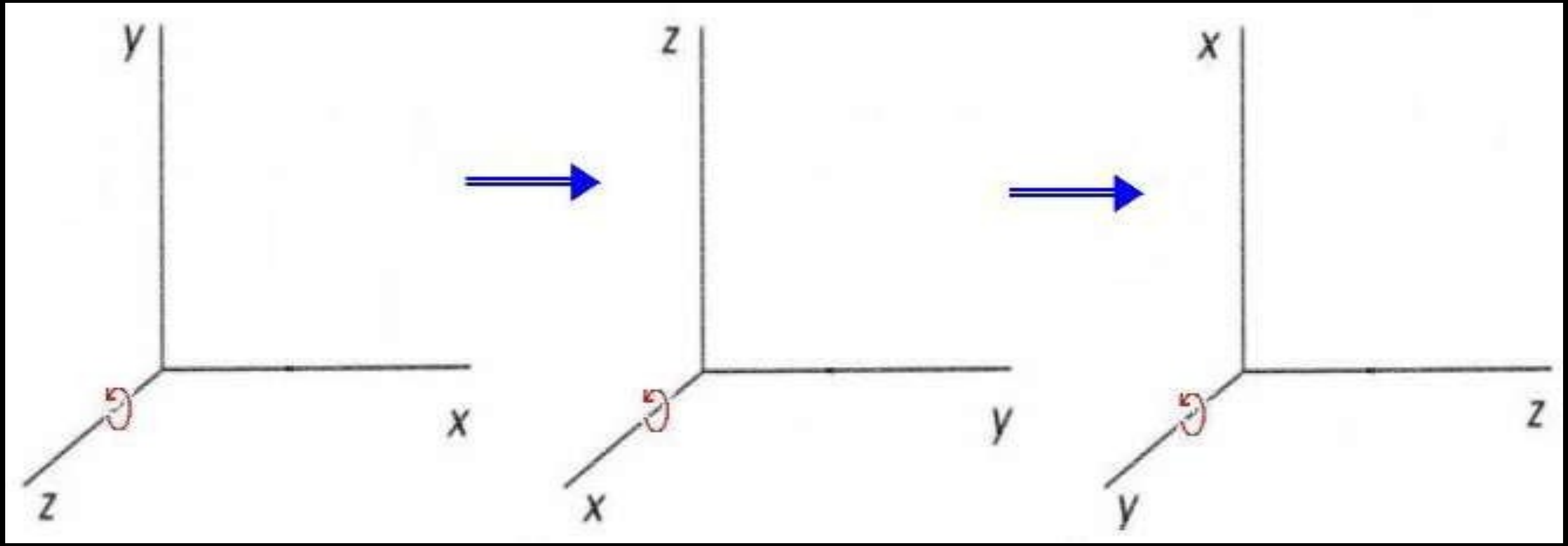
  Z- axis Rotation(Roll) ,Y-axis Rotation(Yaw) , X-axis Rotation(Pitch)



COORDINATE AXIS ROTATION

Obtain rotations around other axes through cyclic permutation of coordinate parameters:

$$x \rightarrow y \rightarrow z \rightarrow x$$



X-AXIS ROTATION

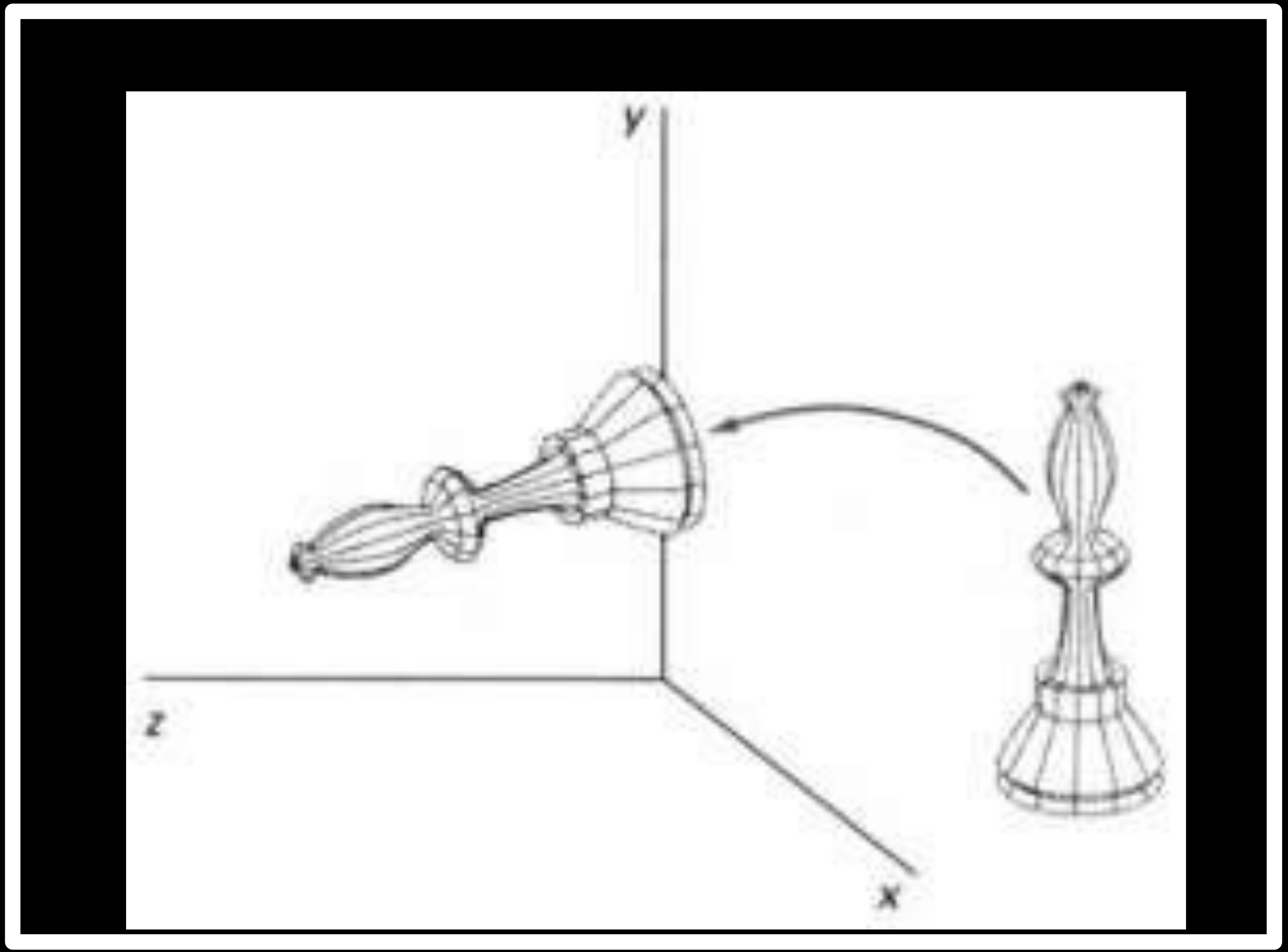
The equation for X-axis rotation

$x' = x$

$y' = y \cos\theta - z \sin\theta$

$z' = y \sin\theta + z \cos\theta$

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



Y-AXISROTATION

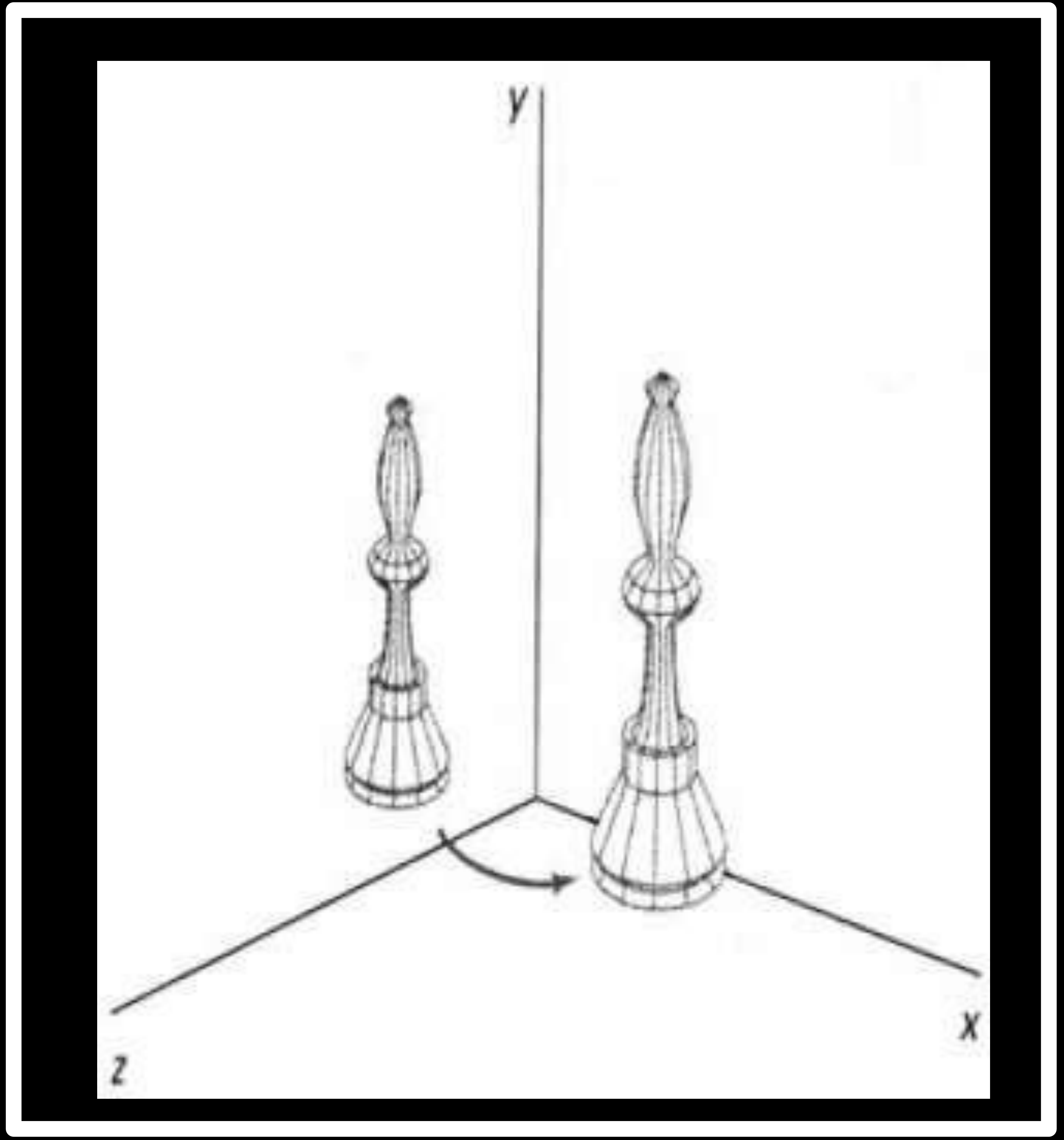
The equation for Y-axis rotaion

$x' = x\cos\theta + z\sin\theta$

$y' = y$

$z' = z\cos\theta - x\sin\theta$

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



Z-AXIS ROTATION

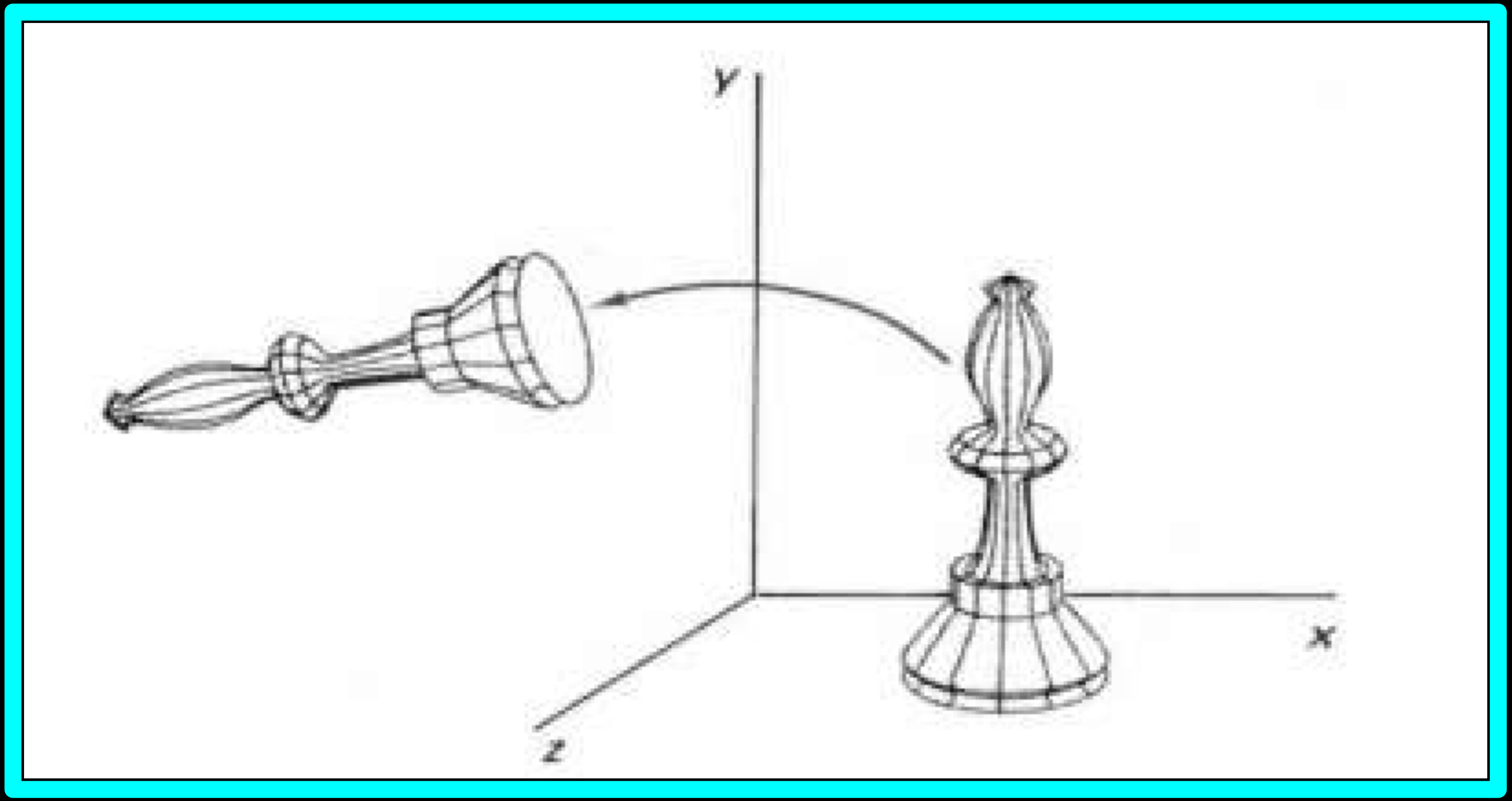
The equation for Z-axis rotaion

$$x' = x\cos\theta - y\sin\theta$$

$$y' = x\sin\theta + y\cos\theta$$

$$z' = z$$

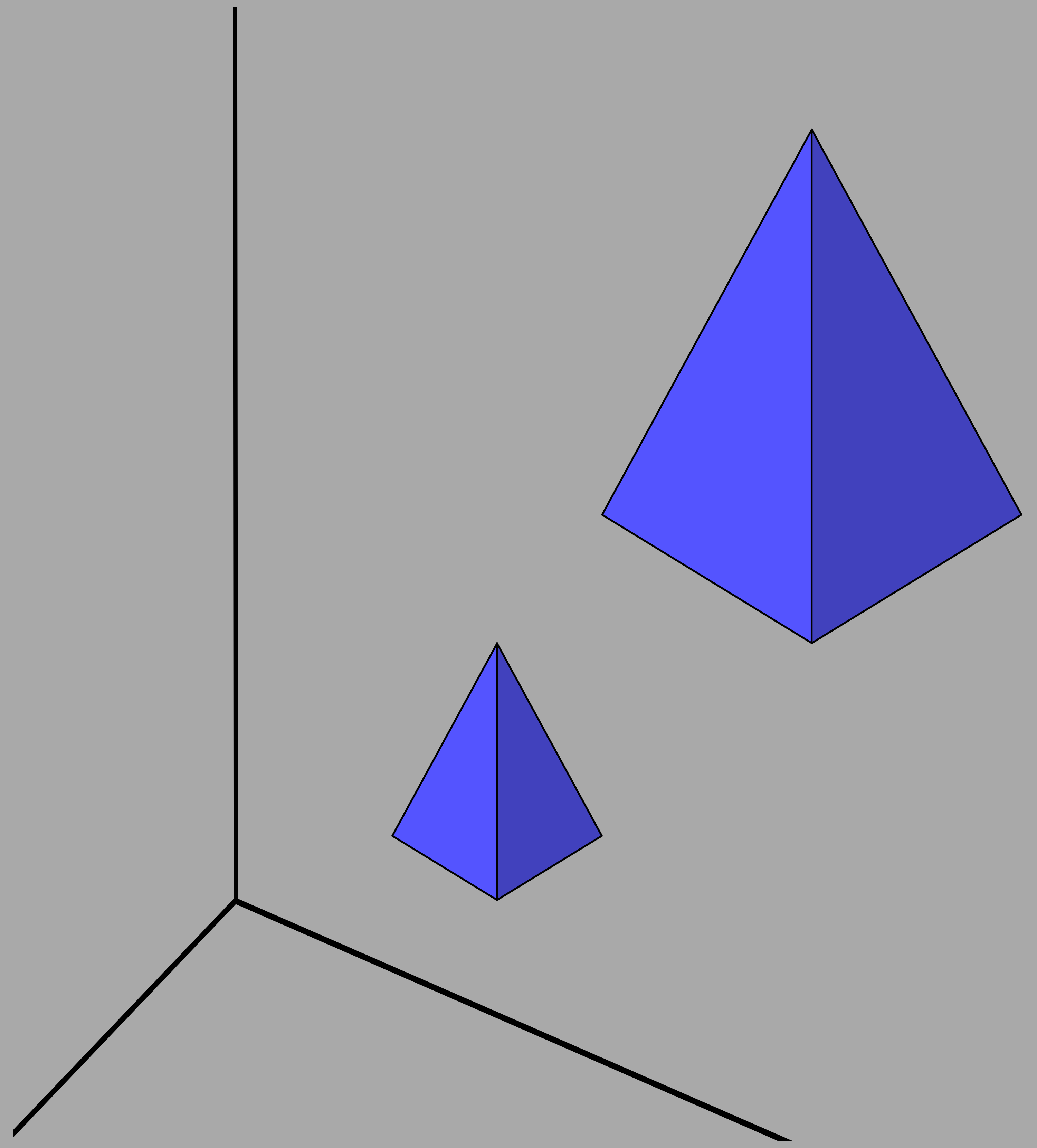
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$



3D SCALING

Changes the size of the object and repositions the object relative to the coordinate origin.

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



3D SCALING

The equations for scaling

$$S_{sx,sy,sz} \quad \begin{aligned} y' &= y \cdot sy \\ z' &= z \cdot sz \\ x' &= x \cdot sx \end{aligned}$$

3D REFLECTION

- Reflection in computer graphics is used to emulate reflective objects like mirrors and shiny surfaces
- Reflection may be an x-axis y-axis , z-axis. and also in the planes xy-plane,yz-plane , and zx-plane.
- Reflection relative to a given Axis are equivalent to 180 Degree rotations

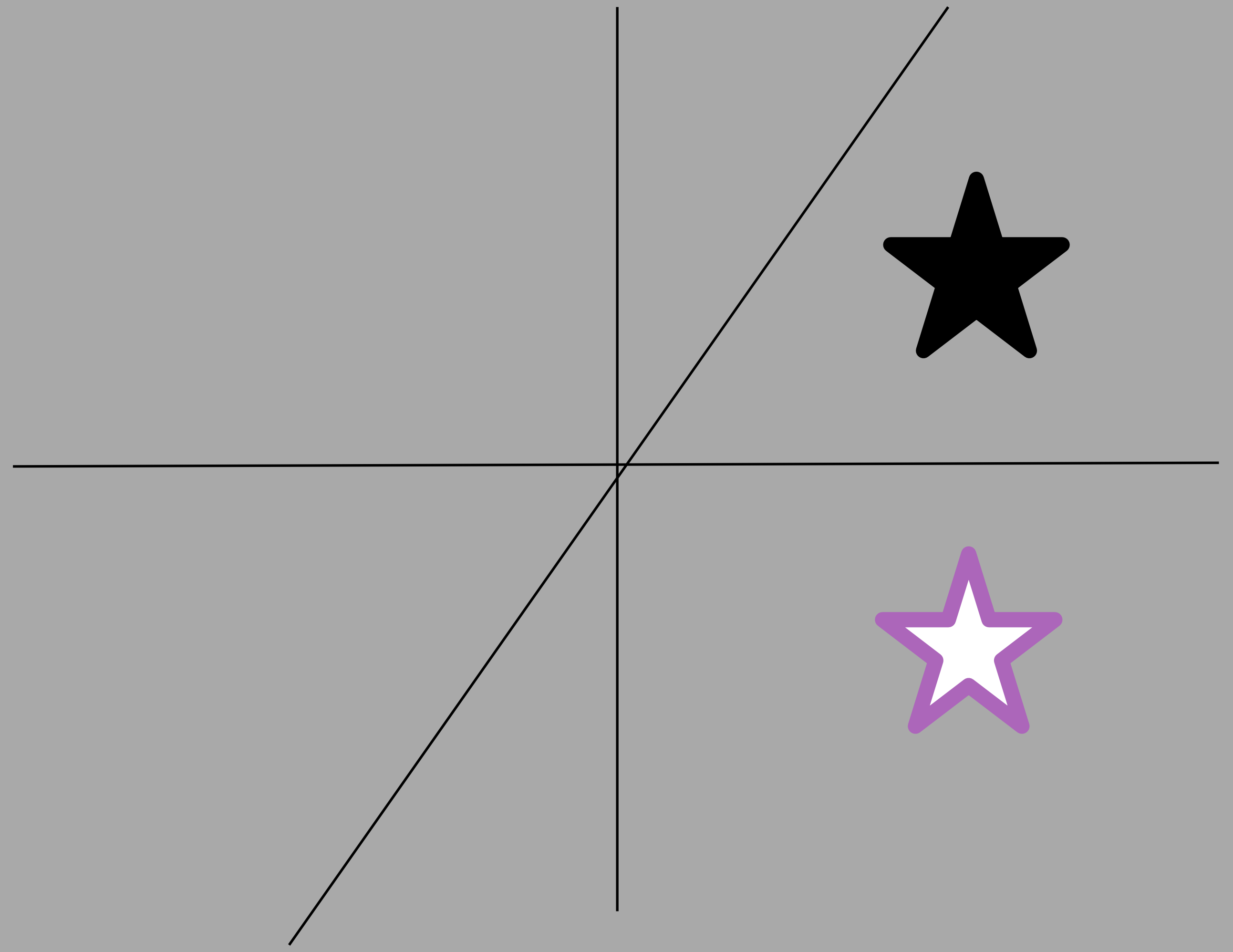
Reflection about x-axis:-

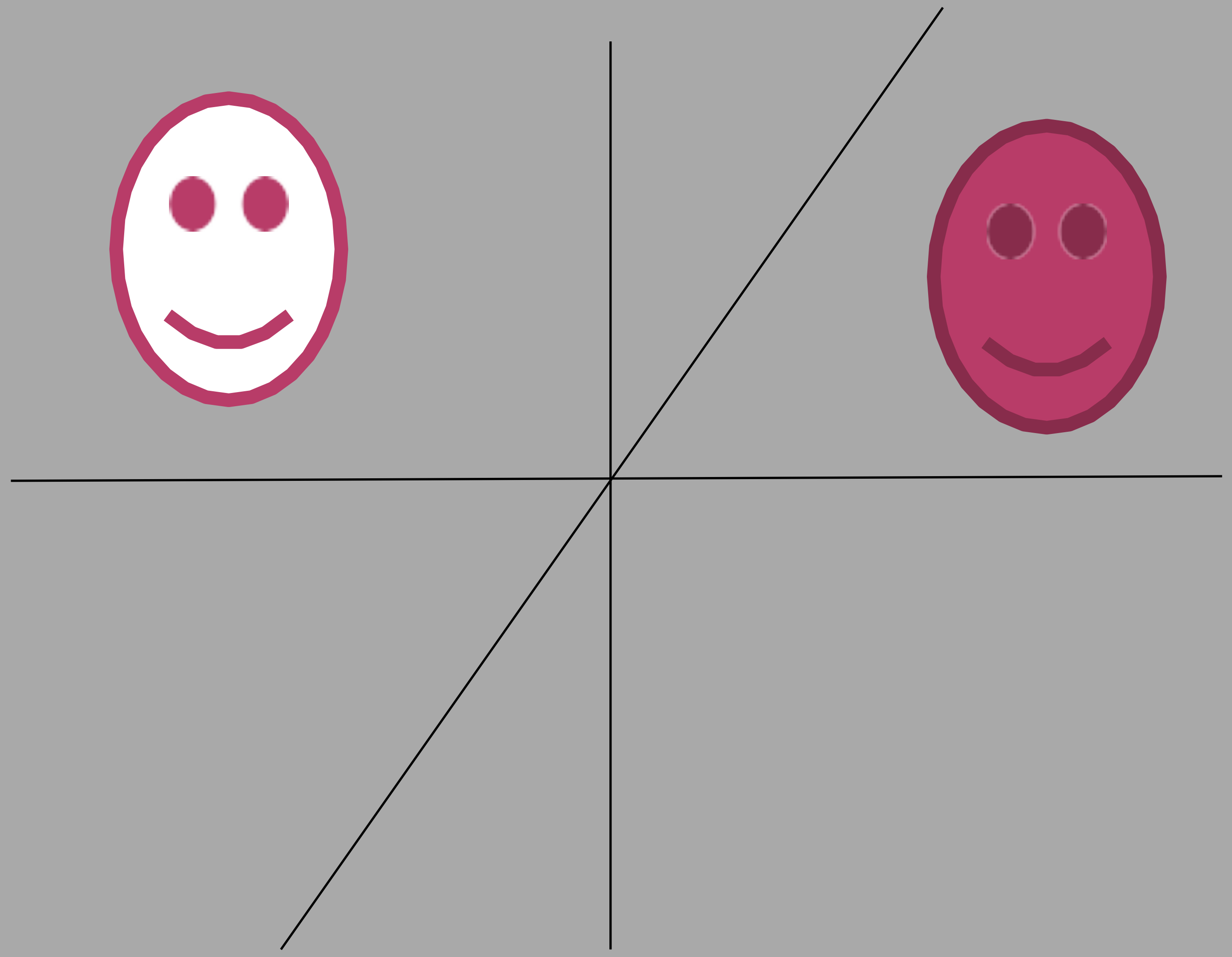
$x'=x$ $y'=-y$ $z'=-z$

1	0	0	0
0	-1	0	0
0	0	-1	0
0	0	0	1

Reflection about y-axis:-

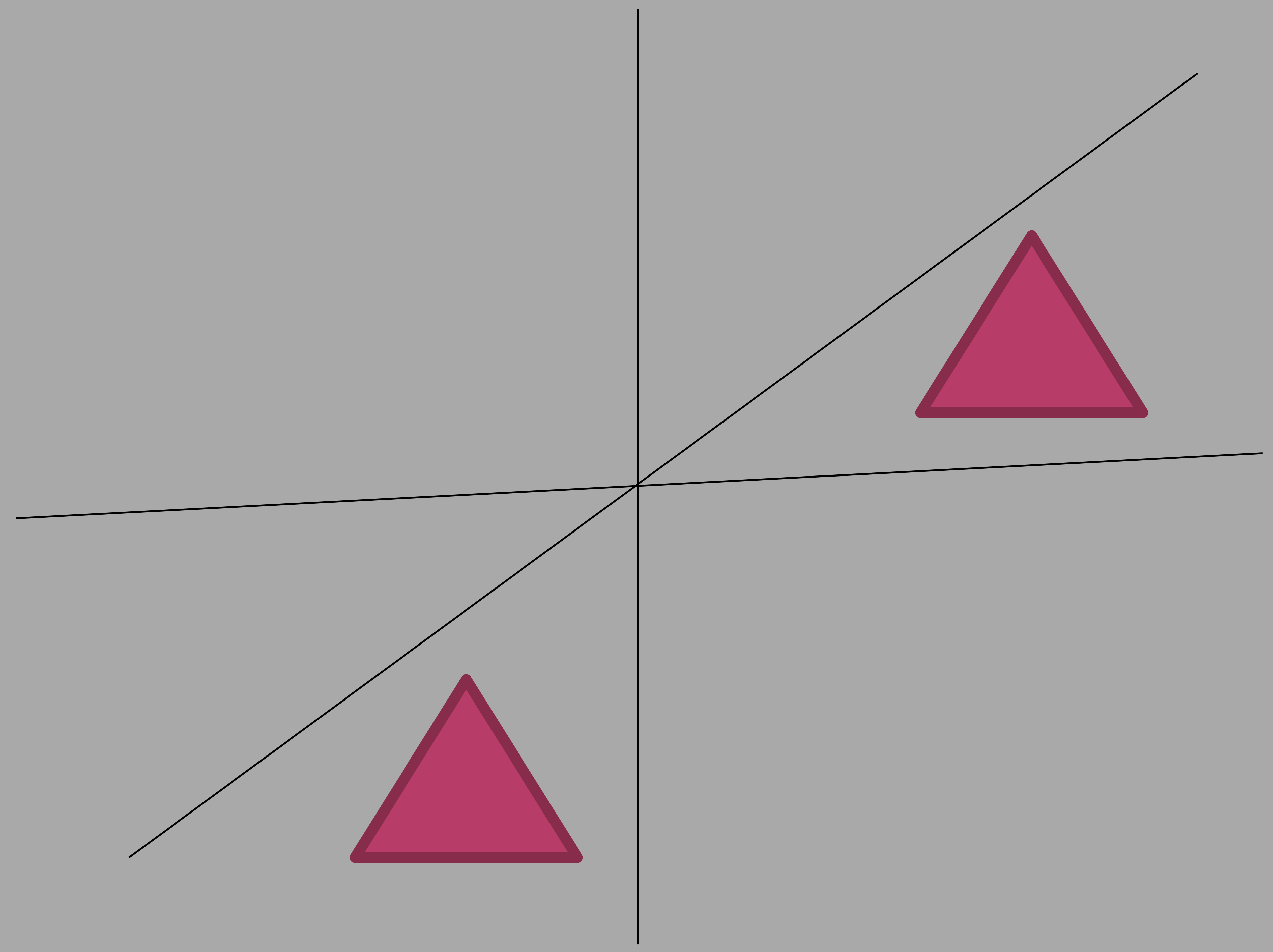
$y'=y$ $x'=-x$ $z'=-z$





The matrix for reflection about y-axis:-

$$\begin{matrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{matrix}$$



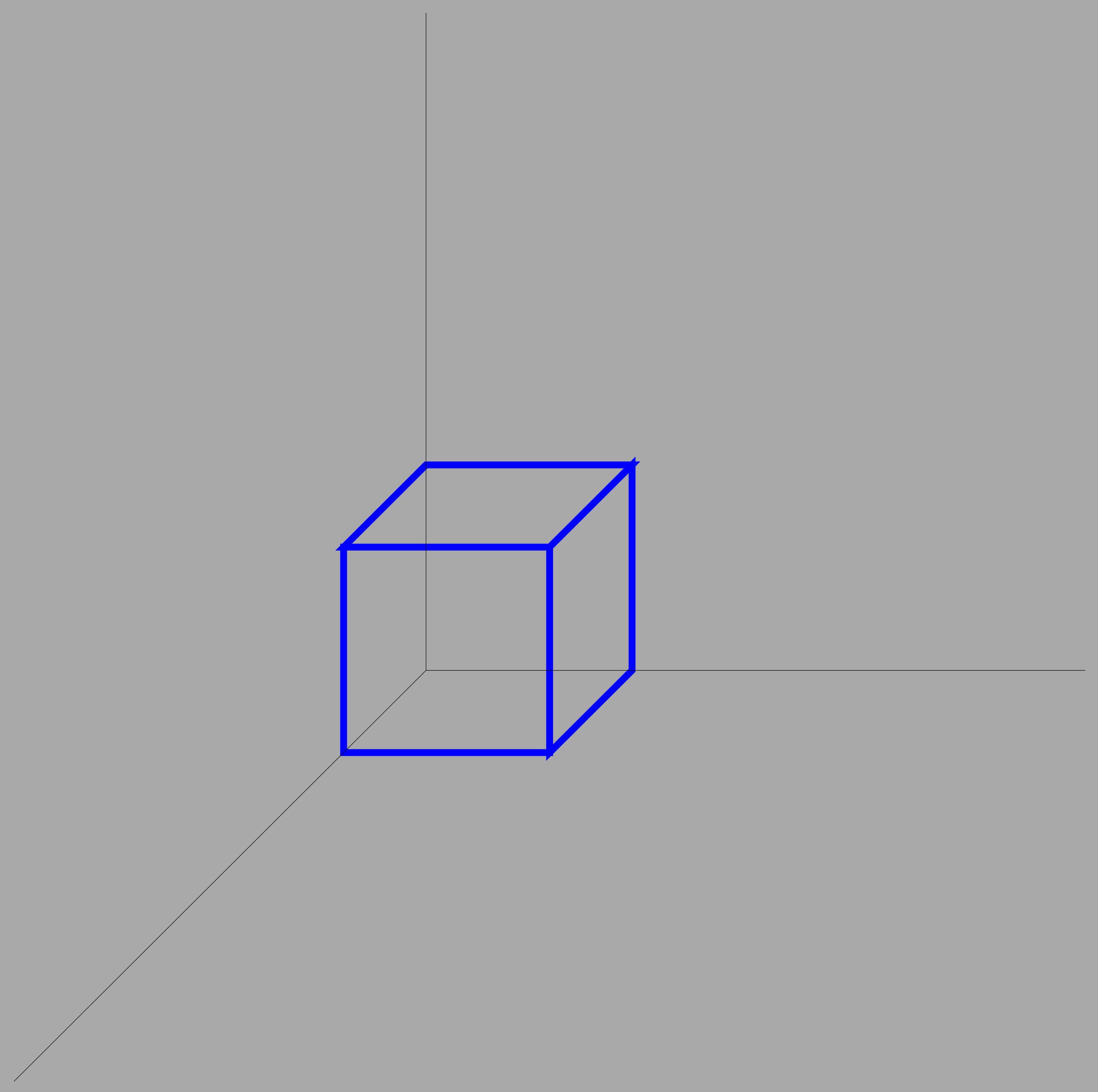
Reflection about z-axis:-

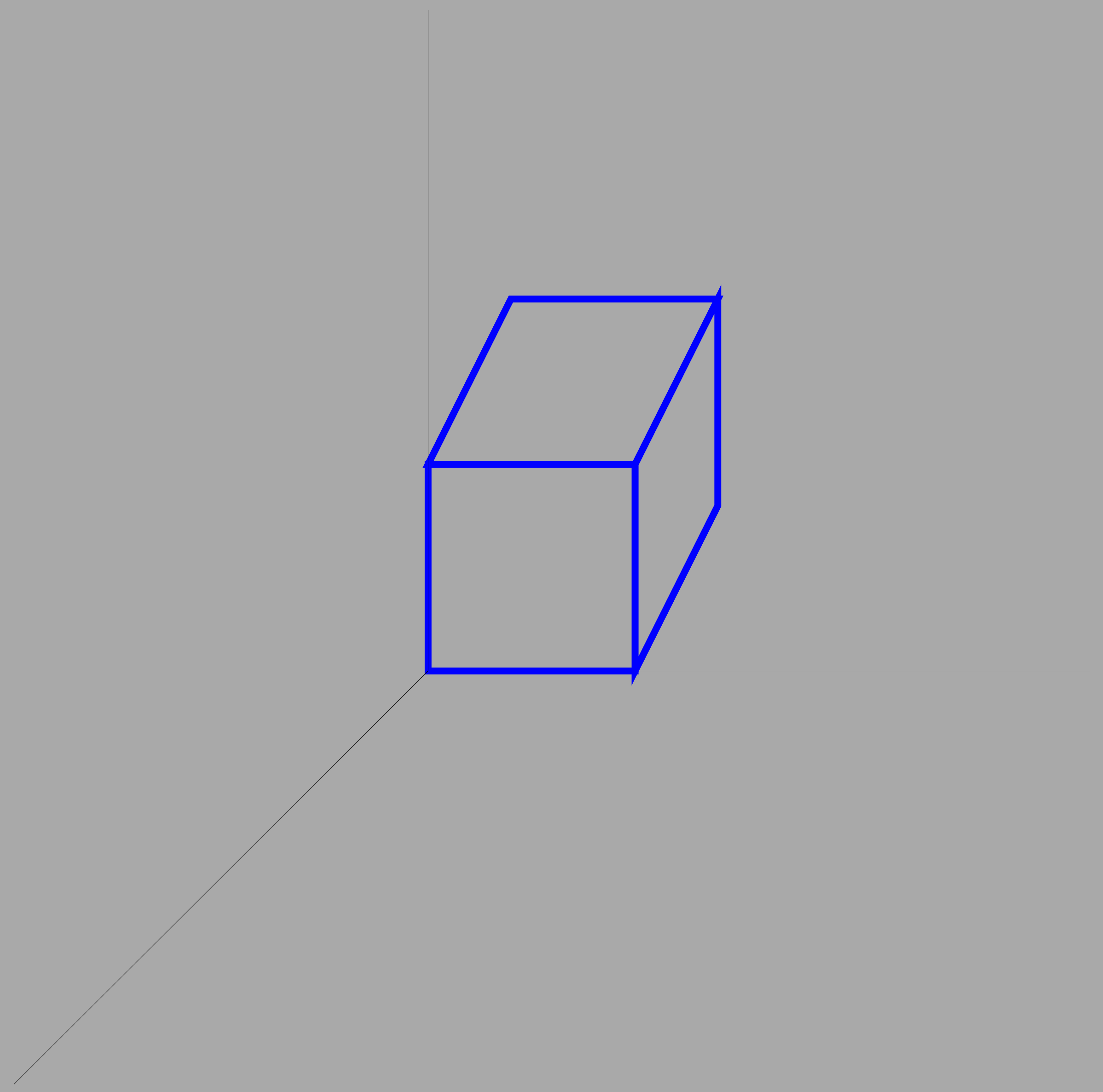
x'=-x   y'=-y   z'=z

3D SHEARING

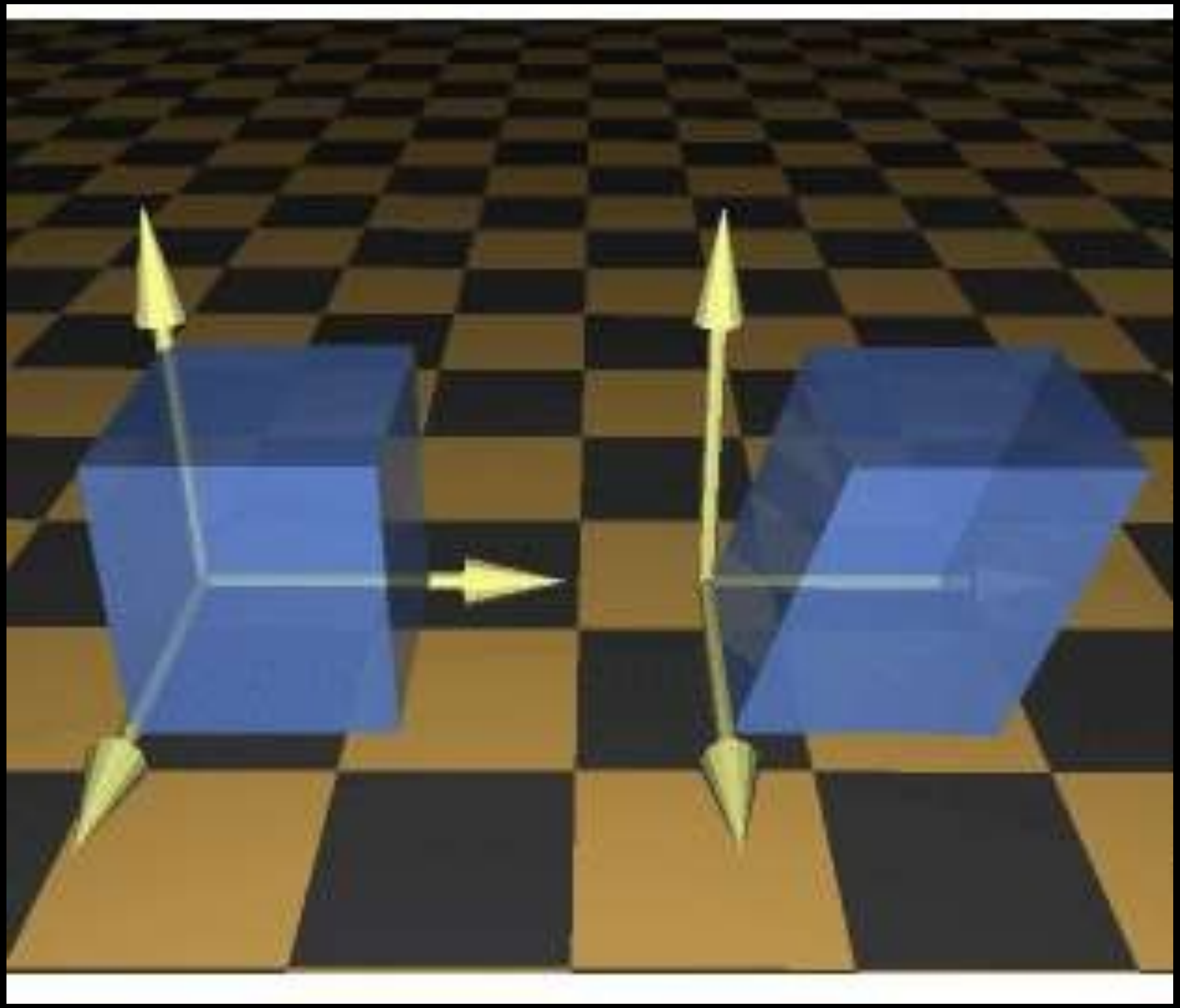
- Modify object shapes
- Useful for perspective projections
- When an object is viewed from different directions and at different distances, the appearance of the object will be different. Such view is called perspective view. Perspective projections mimic what the human eyes see.

E.g. draw a cube (3D) on a screen (2D) Alter the values for **x** and **y** by an amount proportional to the distance from z_{ref}





- Matrix for 3d shearing
- Where a and b can Be assigned any real Value.



$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & a & 0 \\ 0 & 1 & b & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Thank You !!