Affine Transformations

CMSC 161: Interactive Computer Graphics

2nd Semester 2014-2015

Institute of Computer Science

University of the Philippines – Los Baños

Lecture by James Carlo Plaras

Transformation

Function that takes a point or a vector and maps it into another point or vector

$$Q = f(P)$$

$$v = f(u)$$

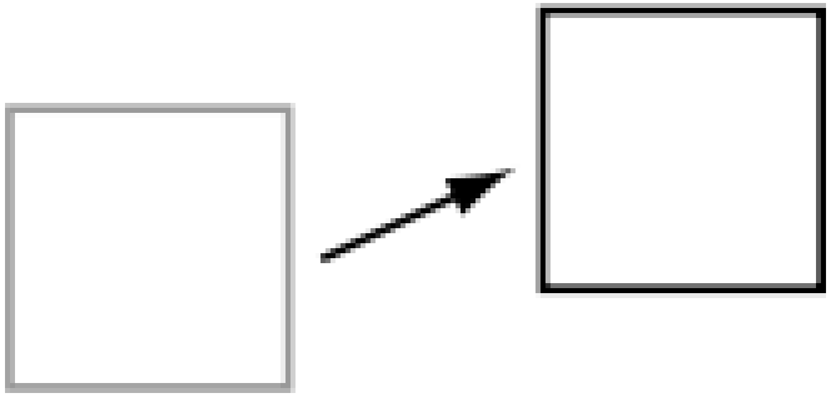
Affine Transformation

Transformations that preserve:

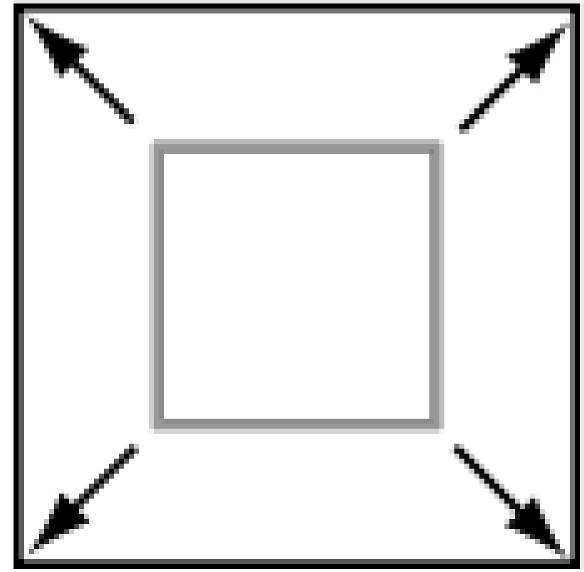
Straight lines

Ratios of distances between collinear points

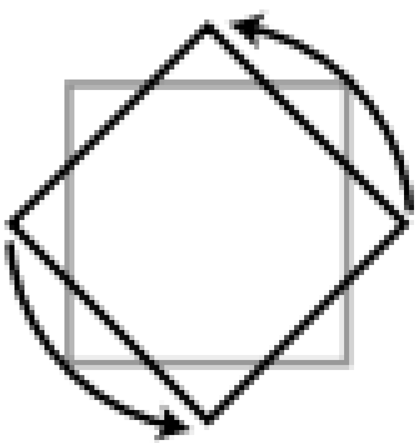
Parallel lines



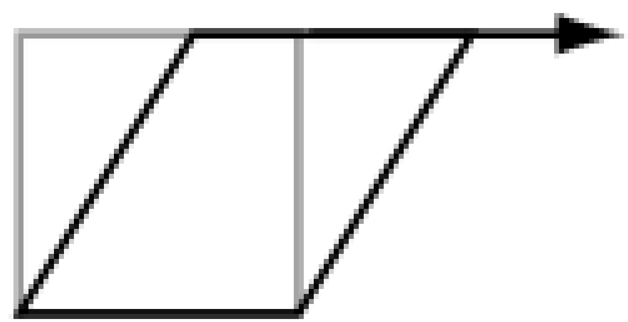
Translate



Scale



Rotate



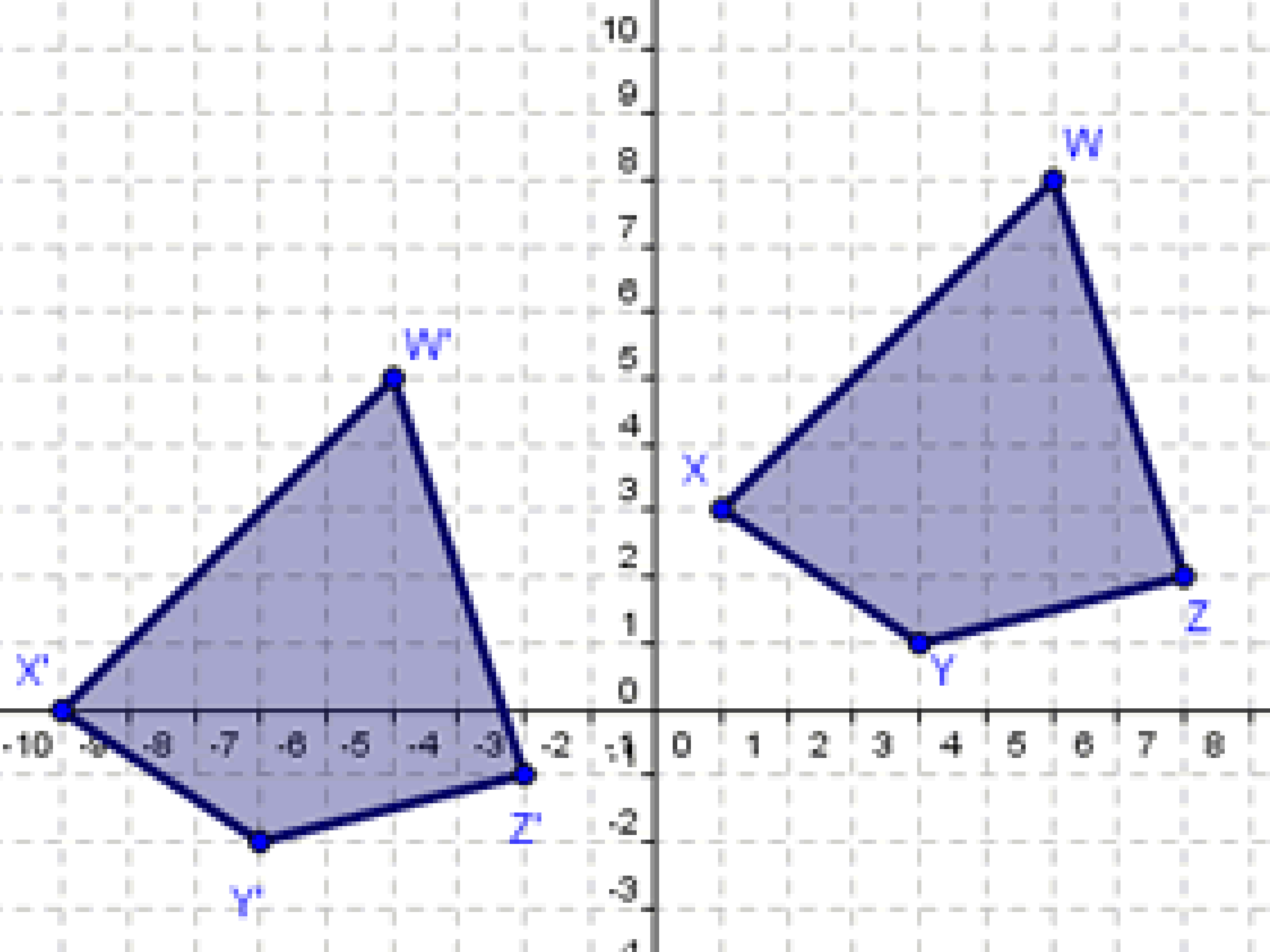
Shear

Translation

Displaces points by a fixed distance in a given direction

Fancy term for movement

$$Q = Translate(P, d)$$


W
X
Y
Z
W'
X'
Y'
Z'
10
9
8
7
6
5
4
3
2
1
0
-1
-2
-3
-10
-8
-7
-6
-5
-4
-3
-2
0
1
2
3
4
5
6
7
8

Translation

$$Q_x = P_x + d_x$$

$$Q_y = P_y + d_y$$

$$Q_z = P_z + d_z$$

Translation (Example)

$$P = (1,2,3)$$
$$d = < 4,4,4 >$$
$$Q = Translate(P, d)$$

Translation (Example)

$$Q_x = P_x + d_x$$

$$Q_y = P_y + d_y$$

$$Q_z = P_z + d_z$$

Translation (Example)

$$Q_x = 1 + 4$$

$$Q_y = 2 + 4$$

$$Q_z = 3 + 4$$

Translation (Example)

$$Q_x = 5$$

$$Q_y = 6$$

$$Q_z = 7$$

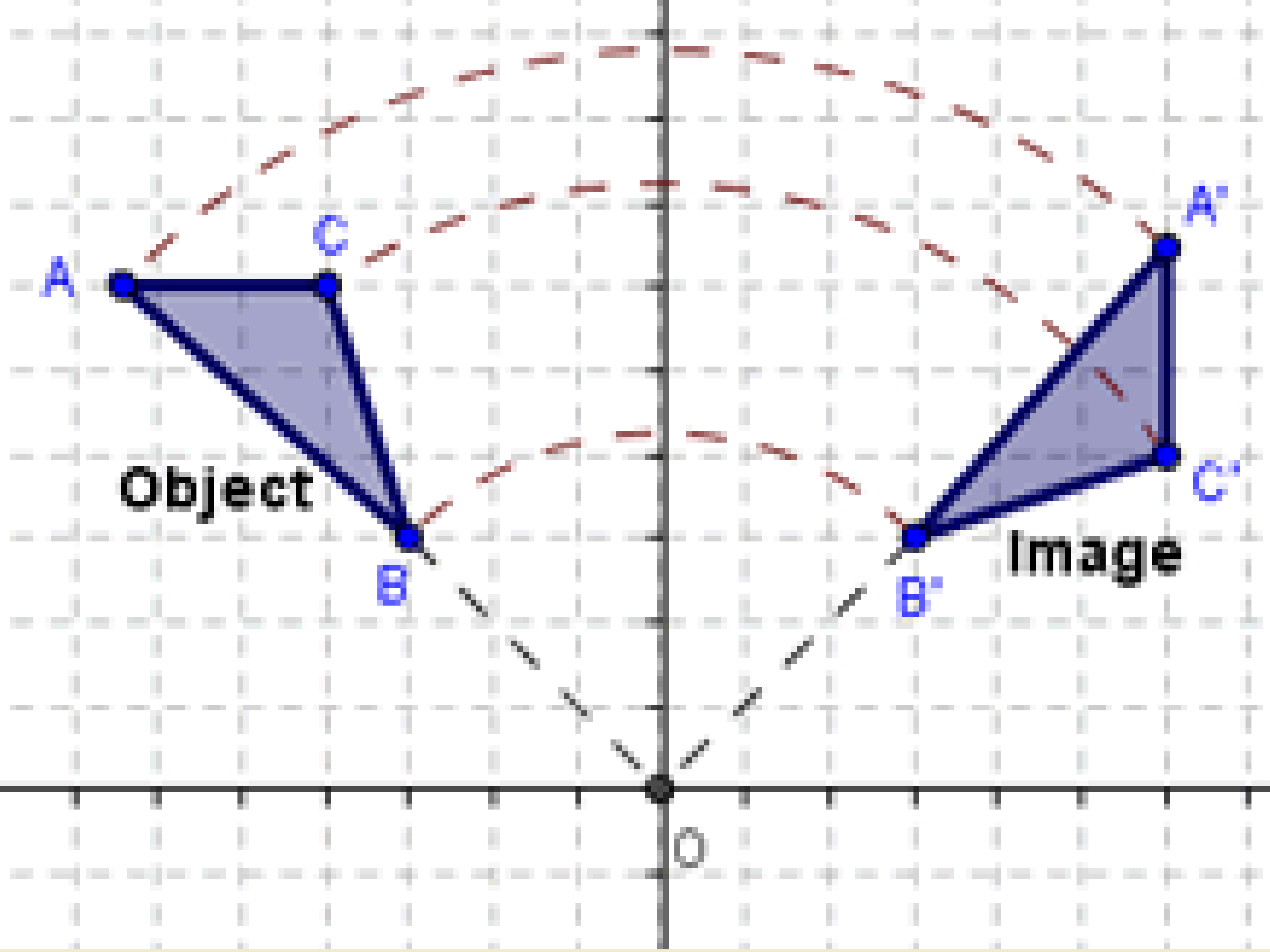
Translation (Example)

$$Q = (5,6,7)$$

Rotation

Re-orienting a given point through some angle

$$Q = Rotate(P, \theta)$$



A
C
B
Object
A'
C'
B'
Image
O

Rotation in 2D (Origin)

$$Q_x = \cos\theta \, P_x - \sin\theta \, P_y$$

$$Q_y = \sin\theta \, P_x + \cos\theta \, P_y$$

$$Q_z = P_z$$

Rotation (Example)

$$P = (1,2,3)$$
$$\theta = 30°$$
$$Q = Rotate(P, d)$$

Rotation (Example)

$$Q_x = \cos\theta\, P_x - \sin\theta\, P_y$$

$$Q_y = \sin\theta\, P_x + \cos\theta\, P_y$$

$$Q_z = P_z$$

Rotation (Example)

$$Q_x = \cos 30° \, 1 - \sin 30° \, 2$$

$$Q_y = \sin 30° \, 1 + \cos 30° \, 2$$

$$Q_z = 3$$

Rotation (Example)

$$Q_x = 0.8660(1) - (0.5000)2$$

$$Q_y = (0.5000)1 + (0.8660)2$$

$$Q_z = 3$$

Rotation (Example)

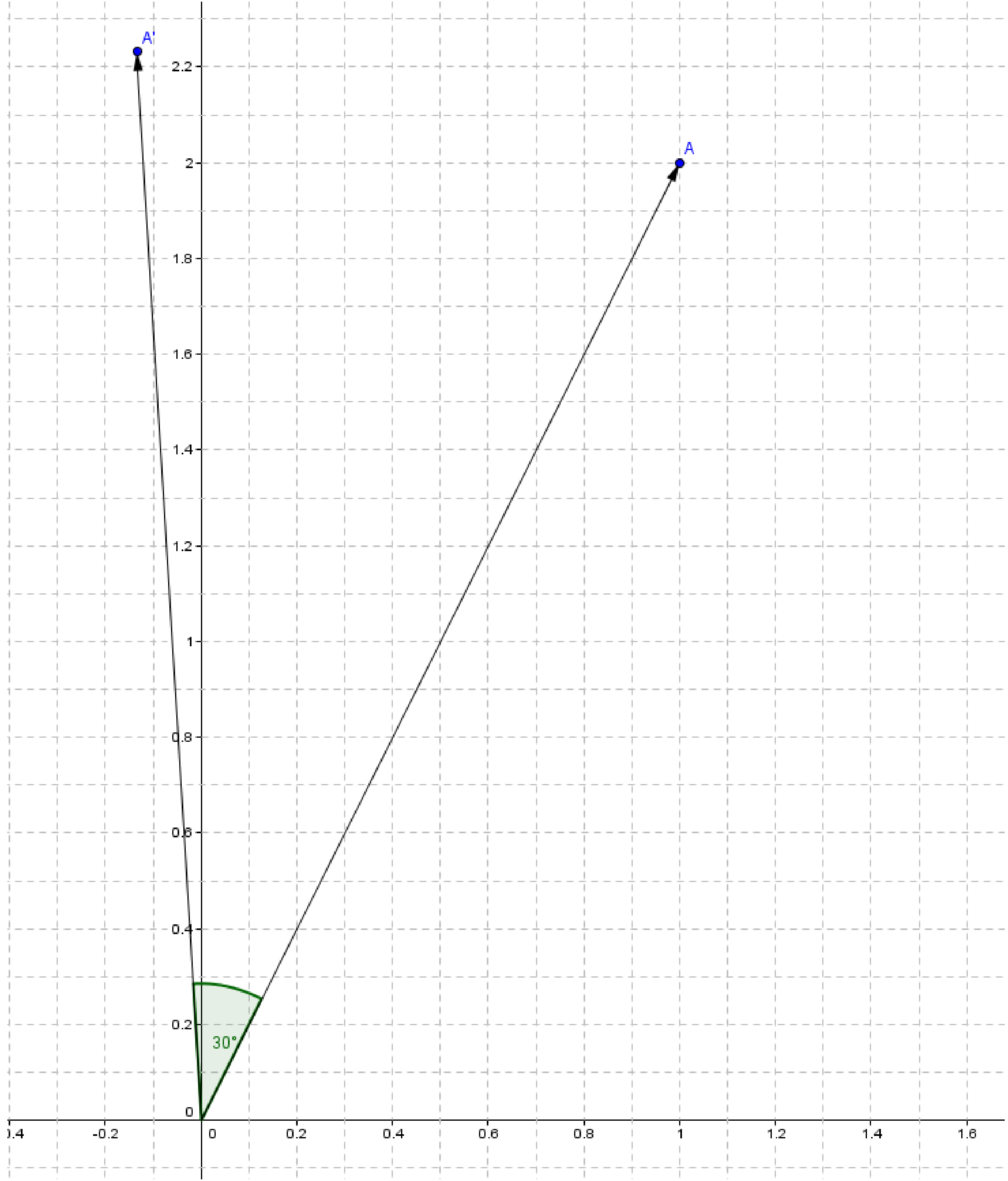
$$Q_x = -0.134$$

$$Q_y = 2.232$$

$$Q_z = 3$$

Rotation (Example)

$$Q = (-0.134, 2.232, 3)$$


A'
A
30°
2.2
2
1.8
1.6
1.4
1.2
1
0.8
0.6
0.4
0.2
0
-0.2
0
0.2
0.4
0.6
0.8
1
1.2
1.4
1.6

Rotation in 3D

With respect to an axis

X-axis rotation

Y-axis rotation

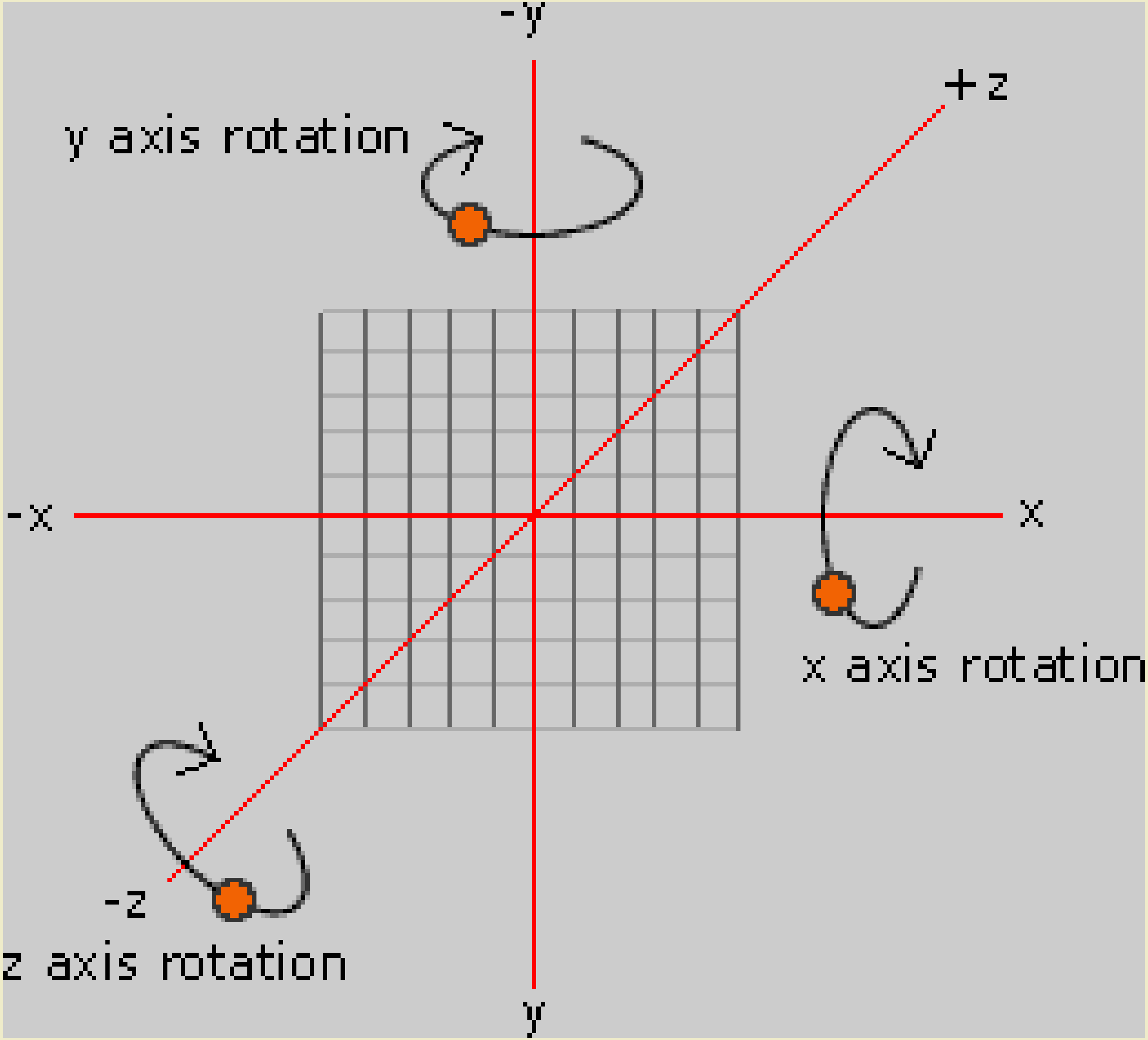
Z-axis rotation

Rotation

Rotation in 2D = Rotation about z-axis

Positive θ means counter-clockwise rotation

Negative θ means clockwise rotation


-y
+z
y axis rotation
-x
x
x axis rotation
-z
z axis rotation
y

Rotation in 3D with respect to x-axis

$$Q_x = P_x$$

$$Q_y = \cos\theta\, P_y - \sin\theta\, P_z$$

$$Q_z = \sin\theta\, P_y + \cos\theta\, P_z$$

Rotation in 3D with respect to y-axis

$$Q_x = \cos\theta\, P_x + \sin\theta\, P_z$$

$$Q_y = P_y$$

$$Q_z = -\sin\theta\, P_x + \cos\theta\, P_z$$

Rotation in 3D with respect to z-axis

$$Q_x = \cos\theta \, P_x - \sin\theta \, P_y$$

$$Q_y = \sin\theta \, P_x + \cos\theta \, P_y$$

$$Q_z = P_z$$

Rigid-Body Transformation

Translation and Rotation is a

Rigid-Body Transformation

It does not alter the shape or volume of the object

Scaling

Non-Rigid-Body Transformation

that alters the size of the object

$$Q = Scale(P, s)$$

Scaling

$$Q_x = s_x P_x$$

$$Q_y = s_y P_y$$

$$Q_z = s_z P_z$$

Scaling (Example)

$$P = (1,2,3)$$
$$s = < 2,3,2 >$$
$$Q = Scale(P, s)$$

Scaling (Example)

$$Q_x = s_x P_x$$

$$Q_y = s_y P_y$$

$$Q_z = s_z P_z$$

Scaling (Example)

$$Q_x = 2(1)$$

$$Q_y = 3(2)$$

$$Q_z = 2(3)$$

Scaling (Example)

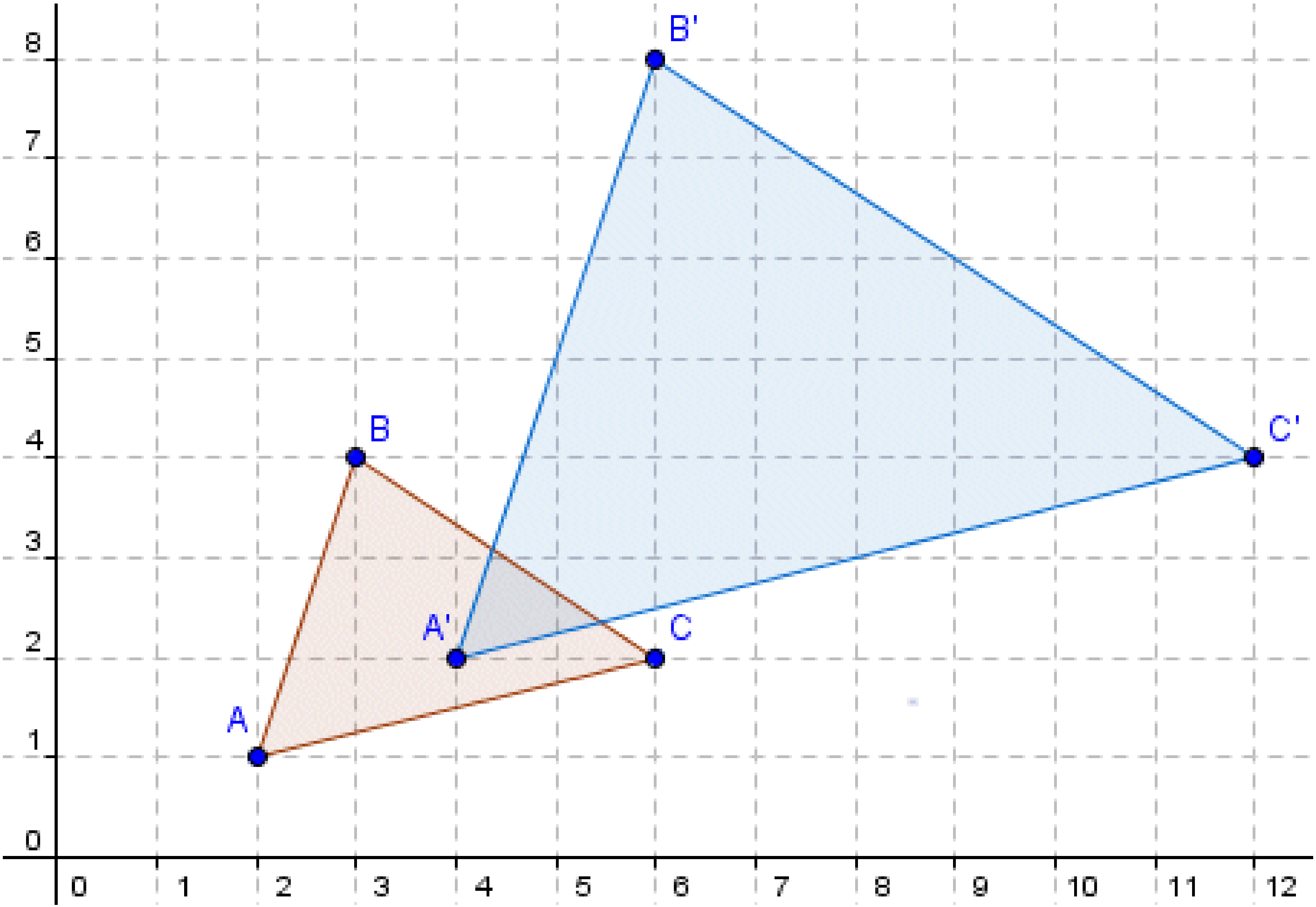
$$Q_x = 2$$

$$Q_y = 6$$

$$Q_z = 6$$

Scaling (Example)

$$Q = (2,6,6)$$



B'
B
C'
A'
C
A
0
1
2
3
4
5
6
7
8
9
10
11
12

LINEAR TRANSFORMATIONS

Linear Transformations

Transformations that respect addition and multiplication

$$f(U + V) = f(U) + f(V)$$

$$f(\alpha V) = \alpha f(V)$$

Linear Transformations

Rotation and Scaling are

Linear Transformations

Translation is a linear transformation only when represented in homogeneous matrix form

Linear Transformations (Example)

$$U = (1,1)$$

$$V = (2,2)$$

$$\theta = 30$$

Linear Transformations (Example)

$$Rotation(U, \theta) + Rotation(V, \theta)$$

$$= Rotation(U + V, \theta)$$

Linear Transformations (Example)

$$(0.37, 1.37) + (0.73, 2.73)$$

$$= Rotation(U + V, \theta)$$

Linear Transformations (Example)

$$(0.37, 1.37) + (0.73, 2.73)$$

$$= Rotation((3,3), 30°)$$

Linear Transformations (Example)

$$(0.37, 1.37) + (0.73, 2.73) = (1.1, 4.1)$$

Linear Transformations (Example)

$$(1.1, 4.1) = (1.1, 4.1)$$

Linear Transformations (Example)

$$U = (1,1)$$

$$V = (2,2)$$

$$d = < 2,2 >$$

Linear Transformations (Example)

$$Translation(U, d) + Translation(V, d)$$

$$= Translation(U + V, d)$$

Linear Transformations (Example)

$$(3,3) + (4,4)$$

$$=$$

$$Translation((3,3), < 2,2 >)$$

Linear Transformations (Example)

$$(3,3) + (4,4) = (5,5)$$

Linear Transformations (Example)

$$(7,7) \neq (5,5)$$

Linear Transformations

Can be represented by a multiplication of a matrix to a vector/point

$$Q = MP$$

Linear Transformations

M is called the transformation matrix

$$Q = MP$$

Linear Transformations

Rotation in 2D (x-axis)

$$Q_x = \cos\theta \, P_x - \sin\theta \, P_y$$

$$Q_y = \sin\theta \, P_x + \cos\theta \, P_y$$

$$Q_z = P_z$$

Linear Transformations

Rotation in 2D (x-axis)

$$Q = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} P$$

Linear Transformations

Rotation in 2D (x-axis)

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \end{bmatrix}$$

$$\boldsymbol{Q} = \boldsymbol{MP}$$

HOMOGENEOUS MATRIX FORM

Translation

$$Q_x = P_x + d_x$$

$$Q_y = P_y + d_y$$

$$Q_z = P_z + d_z$$

$$Q_w = P_w$$

Translation in Homogenous Matrix Form

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ Q_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & d_x \\ 0 & 1 & 0 & d_y \\ 0 & 0 & 1 & d_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ P_w \end{bmatrix}$$

Rotation in 3D with respect to x-axis

$$Q_x = P_x$$

$$Q_y = \cos\theta \, P_y - \sin\theta \, P_z$$

$$Q_z = \sin\theta \, P_y + \cos\theta \, P_z$$

$$Q_w = P_w$$

Rotation (x-axis) in Homogenous Matrix Form

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ Q_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ P_w \end{bmatrix}$$

Rotation in 3D with respect to y-axis

$$Q_x = \cos\theta\, P_x + \sin\theta\, P_z$$

$$Q_y = P_y$$

$$Q_z = -\sin\theta\, P_x + \cos\theta\, P_z$$

$$Q_w = P_w$$

Rotation (y-axis) in Homogenous Matrix Form

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ Q_w \end{bmatrix} = \begin{bmatrix} \cos\theta & 0 & \sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ P_w \end{bmatrix}$$

Rotation in 3D with respect to z-axis

$$Q_x = \cos\theta\, P_x - \sin\theta\, P_y$$

$$Q_y = \sin\theta\, P_x + \cos\theta\, P_y$$

$$Q_z = P_z$$

$$Q_w = P_w$$

Rotation (z-axis)
in Homogenous Matrix Form

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ Q_w \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ P_w \end{bmatrix}$$

Scaling

$$Q_x = s_x P_x$$

$$Q_y = s_y P_y$$

$$Q_z = s_z P_z$$

$$Q_w = P_w$$

Scaling in Homogenous Matrix Form

$$\begin{bmatrix} Q_x \\ Q_y \\ Q_z \\ Q_w \end{bmatrix} = \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ P_w \end{bmatrix}$$

Benefits of Linear Transformations

Allows us to condense operations to a single matrix

Benefits of Linear Transformations

Say we want to do multiple transformations

1. Scale (S)
2. Translate (T)
3. Rotate (R)

$$Q = Rotate\Big(Translate\big(Scale(P)\big)\Big)$$

$$Q = R\Big(H\big(S(P)\big)\Big)$$

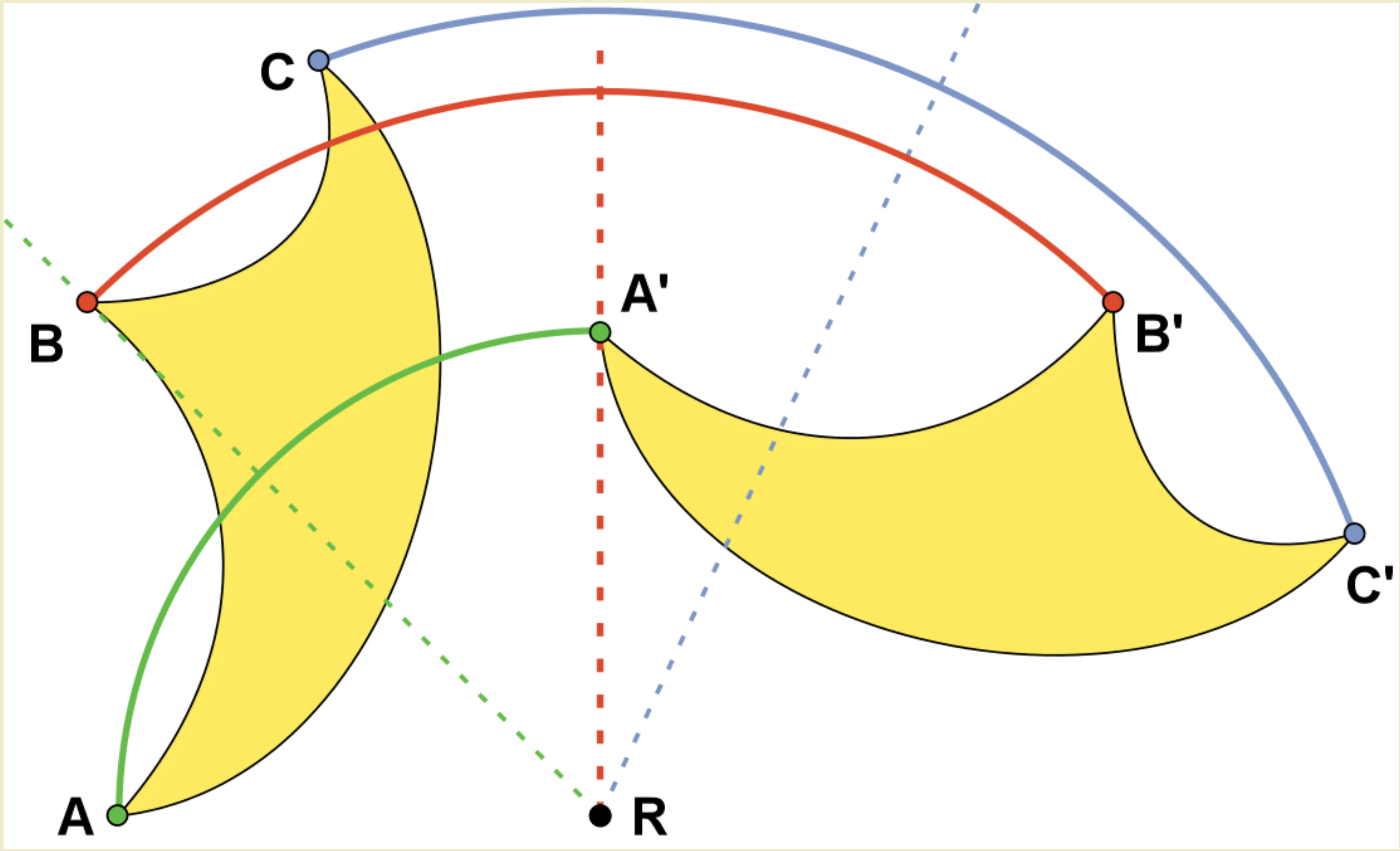
$$Q = RHS\,P$$

$$M = RHS$$

$$\boldsymbol{Q} = \boldsymbol{MP}$$

EXAMPLE COMPLEX TRANSFORMATIONS

Rotation around Arbitrary Point (2D,xy-plane)



Rotation Around an Arbitrary Point in 2D (xy-plane,Homogeneous)

Let X = arbitrary point of rotation

Step 1: Compute t as distance of X from the origin

Step 2: Translate primitive by -t.

Step 3: Rotate result by desired amount.

Step 4: Translate result by t.

Example

$$X = (5,5)$$
$$P = (1,2)$$
$$\theta = 30$$

$$Q = Rotation\ of\ P\ around\ X\ by\ \theta$$

Step 1: Compute T as distance of X from the origin

$$t = X - (0,0)$$
$$t = (5,5) - (0,0)$$
$$t = < 5,5 >$$

Step 2: Translate Primitive by -t

$$Translate(P, -t)$$

$$Translate((1,2), < -5, -5 >$$

$$= (-4, -3)$$

Step 3: Rotate Result by θ

$$
\begin{gathered}
Rotate(P', \theta) \\
= Rotate\big((-4, -3), 30°\big) \\
= (-1.96, -4.6)
\end{gathered}
$$

Step 4: Translate Result by t

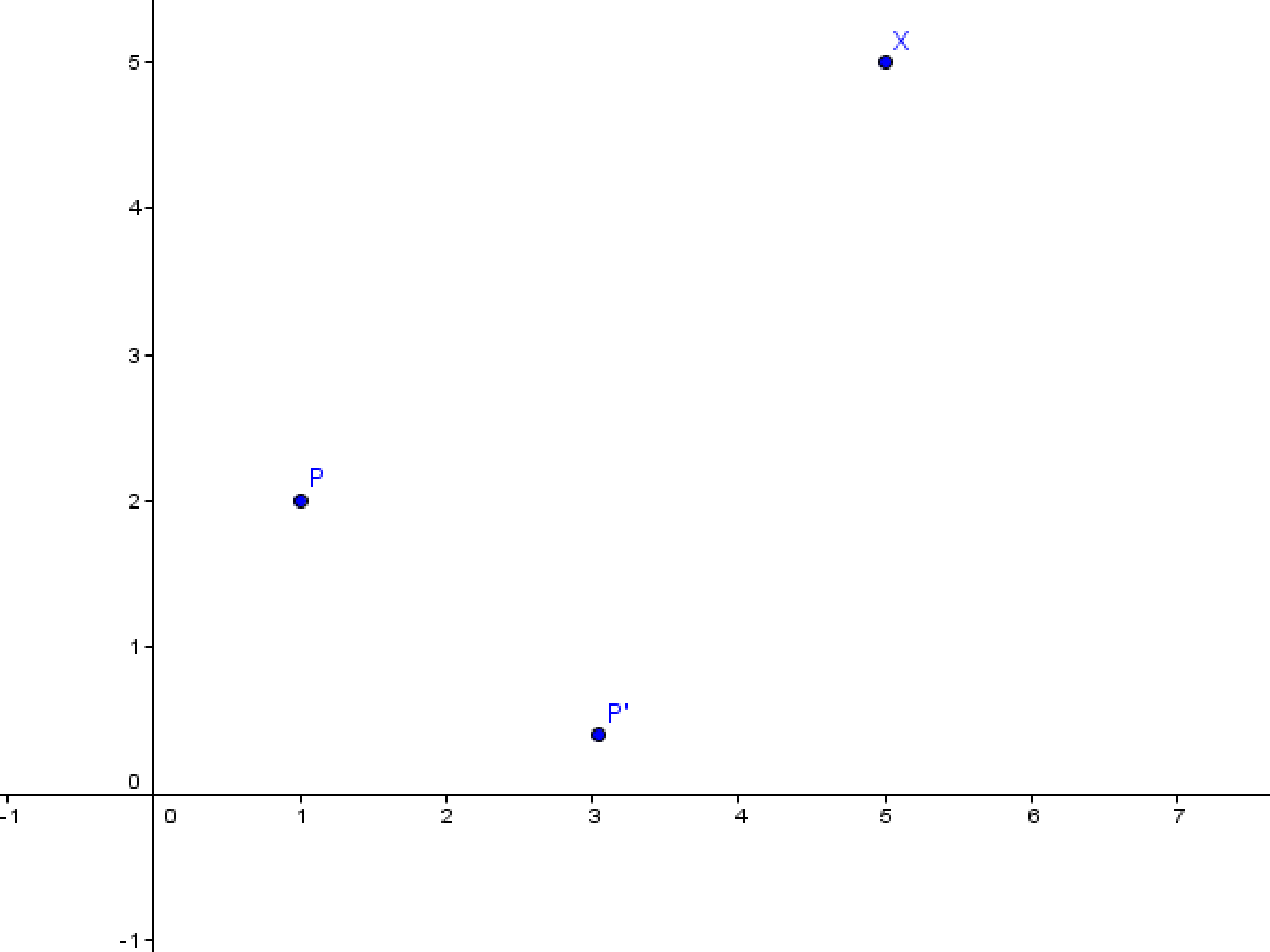
$$Translate(P', t)$$

$$Translate((-1.96, -4.6), < 5,5 >$$

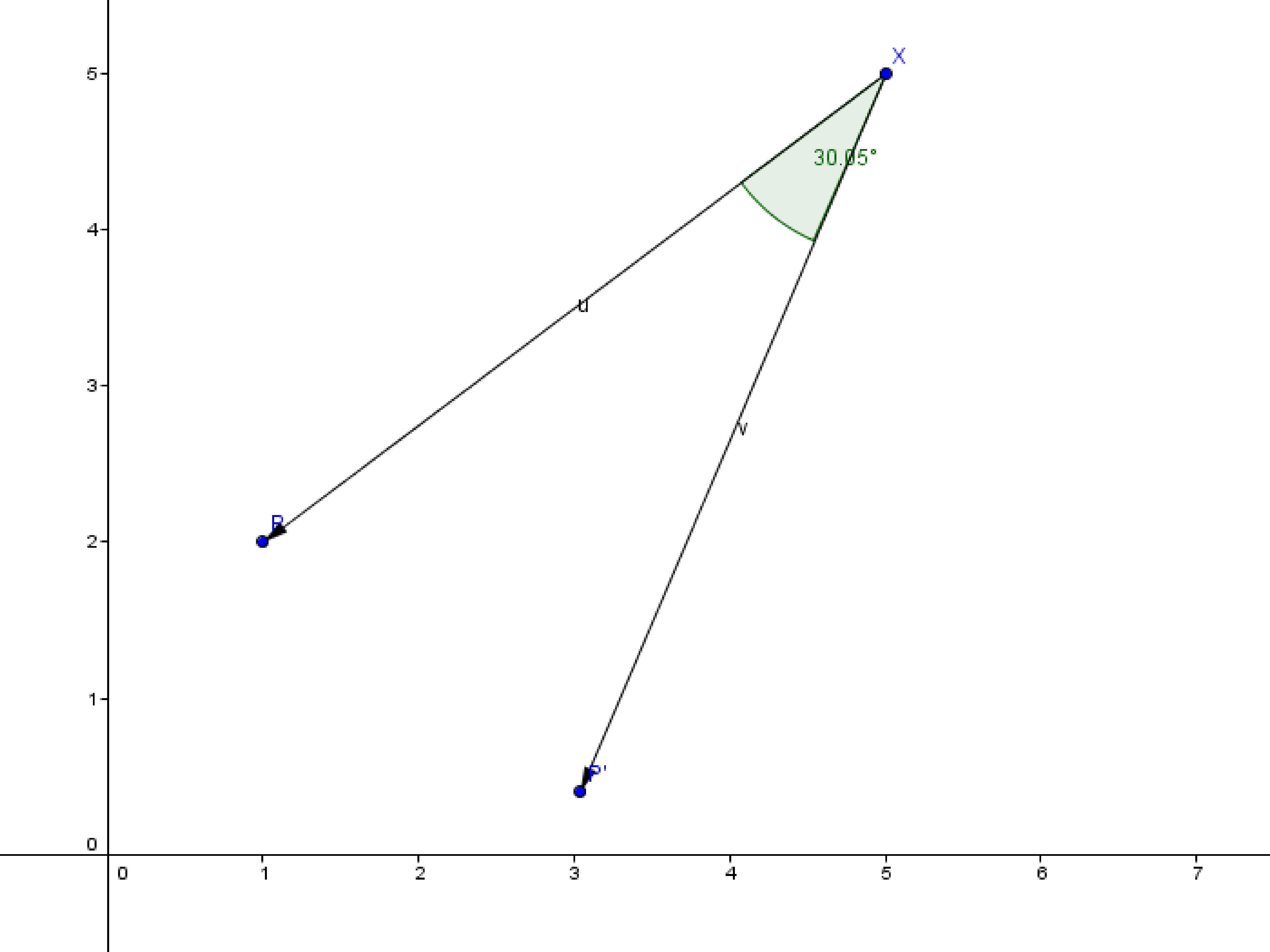
$$= (3.04,0.4)$$

Result

$$P' = (3.04, 0.4)$$



X
P
P'


X
30.05°
u
v
P
P'
0
1
2
3
4
5
6
7
0
1
2
3
4
5

Rotation Around an Arbitrary Point (Homogeneous)

$$P' = Translate(Rotate(Translate(P, -t), \theta\ ), t)$$

$$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ P'_w \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & 0 & -t_x \\ 0 & 1 & 0 & -t_y \\ 0 & 0 & 1 & -t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ P_w \end{bmatrix}$$

Single Transformation Matrix

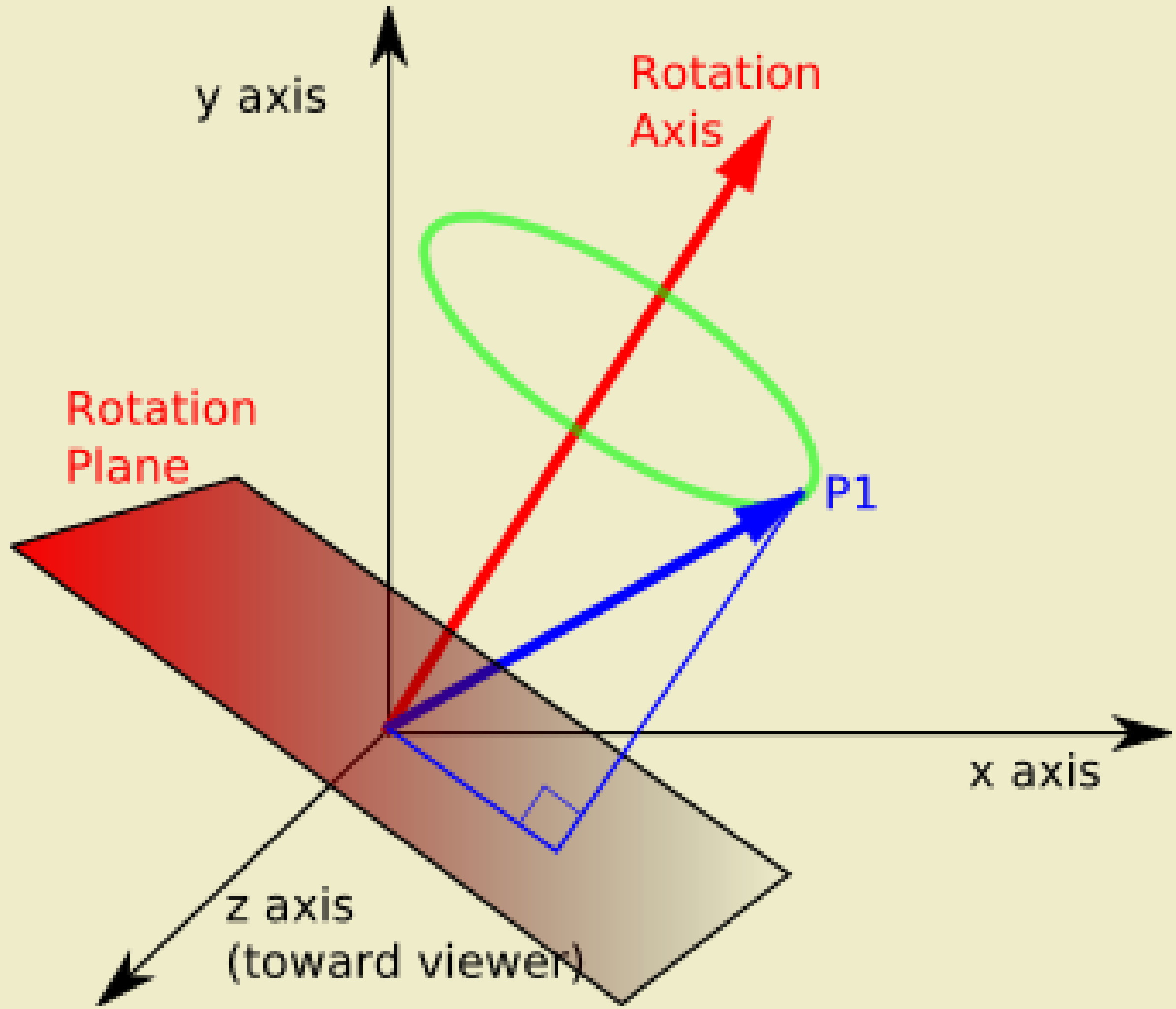
$$\begin{bmatrix} P'_x \\ P'_y \\ P'_z \\ P'_w \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & -T_x\cos\theta + T_y\sin\theta + T_x \\ \sin\theta & \cos\theta & 0 & -T_x\sin\theta - T_y\cos\theta + T_y \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} P_x \\ P_y \\ P_z \\ P_w \end{bmatrix}$$

General Rotations

$$Q = R_z R_y R_x P$$

$$Q = R_x R_y R_z P$$

Rotation around Arbitrary Axis



Rotation around Arbitrary Axis

$$Q = T^{-1} R_x^{-1} R_y^{-1} R_z R_y R_x T P$$

References

Books

- ANGEL, E. AND SHREINER, D. 2012. Interactive computer graphics : a top-down approach with shader-based OpenGL. Addison-Wesley. 6th ed. Boston, MA.

Lecture Slides:

- CLARIÑO, M. CMSC 161 2nd Semester 2011-12 Lecture Slides.
- ALAMBRA, A. CMSC 161 1st Semester 2013-14 Lecture Slides

Images

- http://www.emathematics.net/imagenes/rotation2.gif
- http://www.emathematics.net/imagenes/traslacion5.gif
- http://www.technologyuk.net/mathematics/geometry/images/geometry_0128.gif
- http://www.cs.berkeley.edu/~sequin/CS184/TOPICS/Kinematics/Rot-Compound_2D.GIF
- http://www.euclideanspace.com/maths/algebra/realNormedAlgebra/quaternions/geometric/axisAngle/axisAngle1.png