Geometric Objects and Transformation

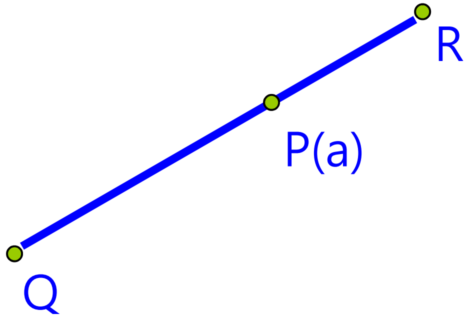
527970
Fall 2020
10/15/2020

Kyoung Shin Park
Computer Engineering
Dankook University

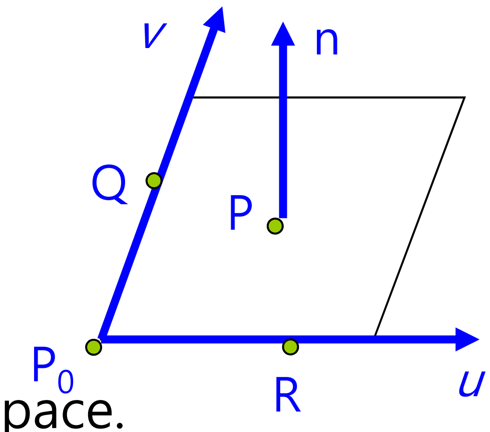
Geometric Objects

- Line
  - 2 points: R, Q
  - v = R- Q

    $P = Q + \alpha v = Q + \alpha(R - Q) = \alpha R + (1 - \alpha)Q$
  - $P = \alpha_1 R + \alpha_2 Q$ where $\alpha_1 + \alpha_2 = 1$ (affine sum)



- Plane
  - 3 points: P_0, Q, R
  - $T(\alpha, \beta) = P + \alpha u + \beta v$
  - $n \bullet (P - P_0) = 0$ where $n = u \times v$



- 3D objects
  - It is a set of vertices in three dimensional space.
  - It is described by the surface, and is hollow.
  - It can be composed of convex polygons.
  - An arbitrary polygon is divided into triangular polygons, i.e., tessellate.

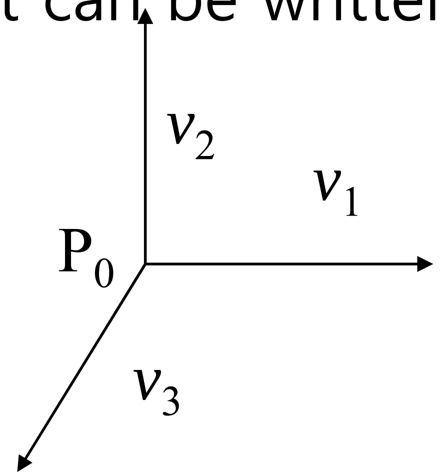
Coordinate Systems

- Consider a basis, v_1, v_2,...., v_n
- Any vector v can be written as $v = a_1 v_1 + a_2 v_2 + ... + a_n v_n$
- The list of scalars $\{a_1, a_2, ... a_n\}$ is the representation of v with respect to the given basis.
- We can write the representation as a row or column array of scalars.

$$\mathbf{a} = [\alpha_1 \;\; \alpha_2 \;\; .... \;\; \alpha_n]^T = \begin{bmatrix} \alpha_1 \\ \alpha_2 \\ . \\ \alpha_n \end{bmatrix}$$

Frames

- The affine space contains points.
- If we work in an affine space we can add the origin to the basis vectors to form a **frame**.
- Frame: (P_0, v_1, v_2, v_3)
- Within this frame, every vector can be written as:
  $v = \alpha_1 v_1 + \alpha_2 v_2 + \ldots + \alpha_n v_n$
- Every point can be written as: $P = P_0 + \beta_1 v_1 + \beta_2 v_2 + \ldots + \beta_n v_n$



$\mathbf{v} = [\alpha_1\ \alpha_2\ \alpha_3\ 0]^T$

$\mathbf{p} = [\beta_1\ \beta_2\ \beta_3\ 1]^T$

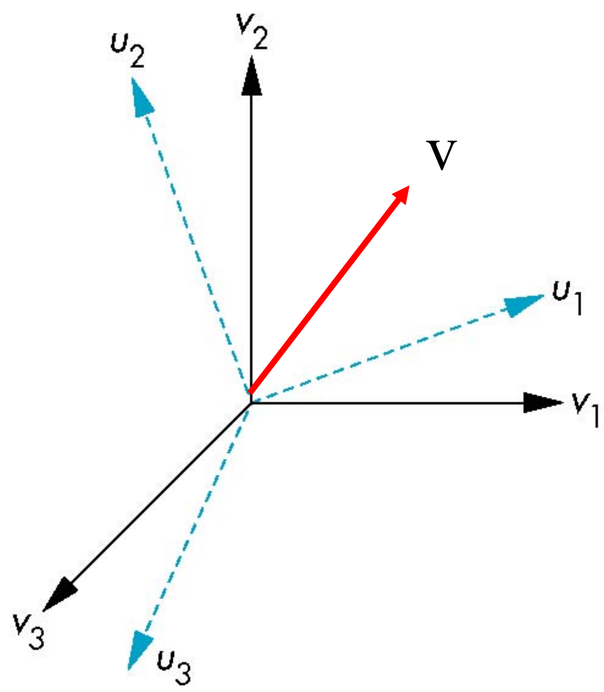
Change of Coordinate Systems

- Consider two representations of a the same vector, v, with respect to two different bases : $\{v_1, v_2, v_3\}$, $\{u_1, u_2, u_3\}$

$$u_1 = \gamma_{11}v_1 + \gamma_{12}v_2 + \gamma_{13}v_3$$
$$u_2 = \gamma_{21}v_1 + \gamma_{22}v_2 + \gamma_{23}v_3$$
$$u_3 = \gamma_{31}v_1 + \gamma_{32}v_2 + \gamma_{33}v_3$$

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} \\ \gamma_{21} & \gamma_{22} & \gamma_{23} \\ \gamma_{31} & \gamma_{32} & \gamma_{33} \end{bmatrix}$$

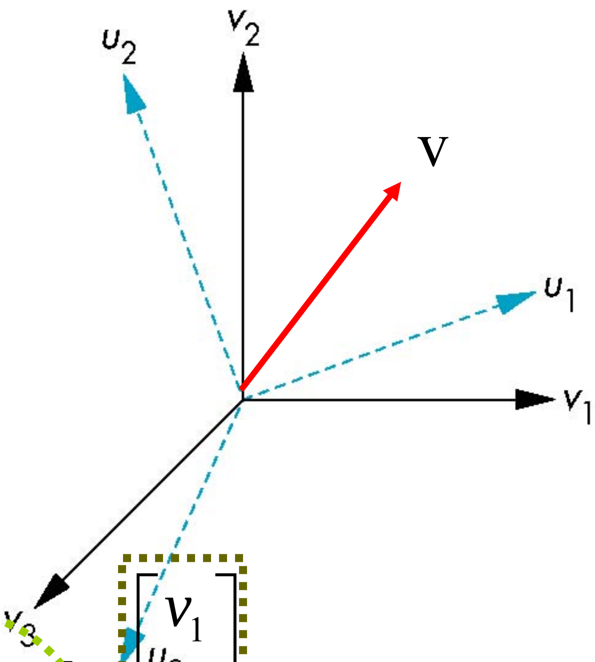
$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \end{bmatrix}$$



Change of Coordinate Systems

$$v = \alpha_1 v_1 + \alpha_2 v_2 + \alpha_3 v_3 = \underbrace{\begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3\end{bmatrix}}_{a^T}\begin{bmatrix}v_1\\ v_2\\ v_3\end{bmatrix}$$

$$v = \underbrace{\begin{bmatrix}\beta_1 & \beta_2 & \beta_3\end{bmatrix}}_{b^T}\begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix}$$

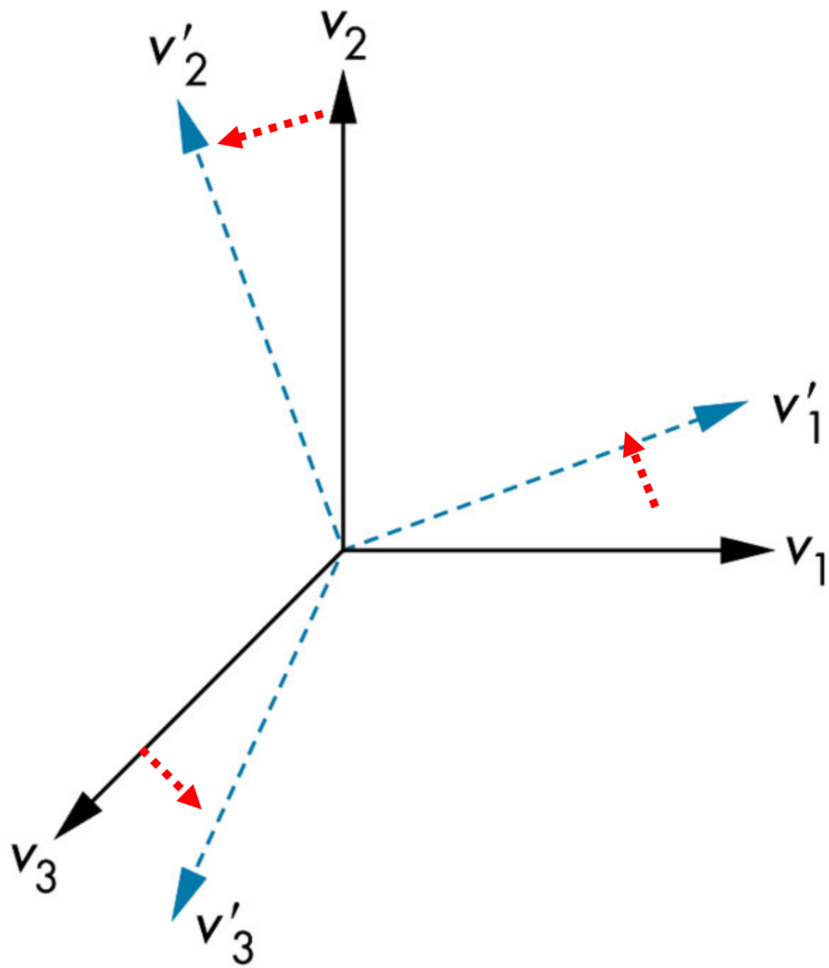


$$\begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3\end{bmatrix}\begin{bmatrix}v_1\\ v_2\\ v_3\end{bmatrix} = \begin{bmatrix}\beta_1 & \beta_2 & \beta_3\end{bmatrix}\begin{bmatrix}u_1\\ u_2\\ u_3\end{bmatrix} = \begin{bmatrix}\beta_1 & \beta_2 & \beta_3\end{bmatrix} M \begin{bmatrix}v_1\\ v_2\\ v_3\end{bmatrix}$$

$$\underbrace{\begin{bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\end{bmatrix}}_{a} = M^T \underbrace{\begin{bmatrix}\beta_1\\ \beta_2\\ \beta_3\end{bmatrix}}_{b} \quad \therefore \mathrm{a} = \mathrm{M}^T \mathrm{b} \quad \therefore \mathrm{b} = (\mathrm{M}^T)^{-1}\mathrm{a}$$

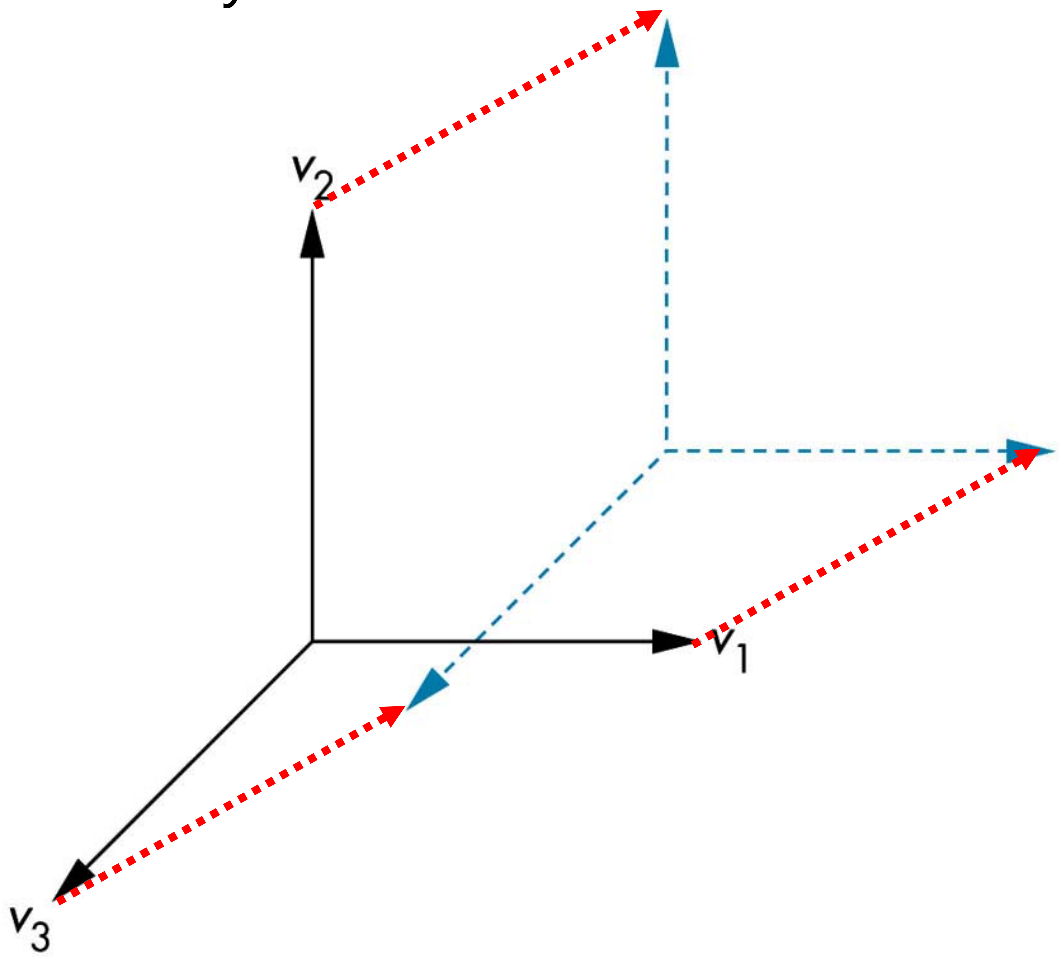
Rotation and Scaling of a Basis

- The rotation and scaling transformation can be represented by the basis vectors.



Translation of a Basis

- However, a simple translation of the origin is not represented by a 3x3 matrix.

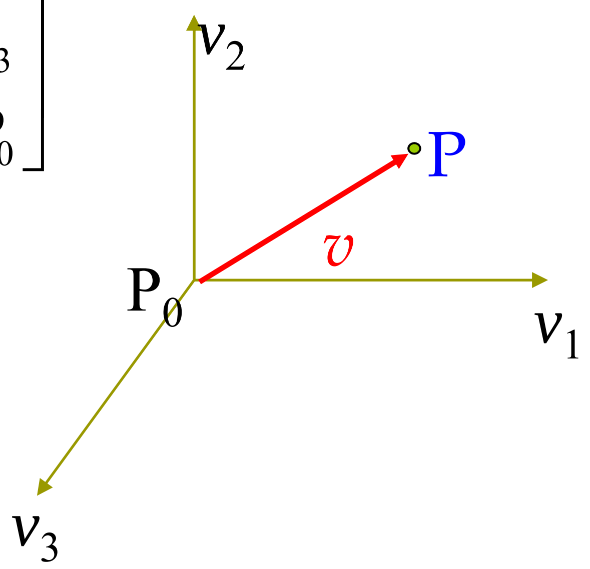


Homogeneous Coordinates

$$\textbf{vector } v = \sum \alpha_i v_i = \begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3 & 0\end{bmatrix}\begin{bmatrix}v_1\\ v_2\\ v_3\\ P_0\end{bmatrix}$$

$$\textbf{point } P = P_0 + \sum \alpha_i v_i = \begin{bmatrix}\alpha_1 & \alpha_2 & \alpha_3 & 1\end{bmatrix}\begin{bmatrix}v_1\\ v_2\\ v_3\\ P_0\end{bmatrix}$$

$$P = \begin{bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\\ 1\end{bmatrix}, v = \begin{bmatrix}\alpha_1\\ \alpha_2\\ \alpha_3\\ 0\end{bmatrix}$$



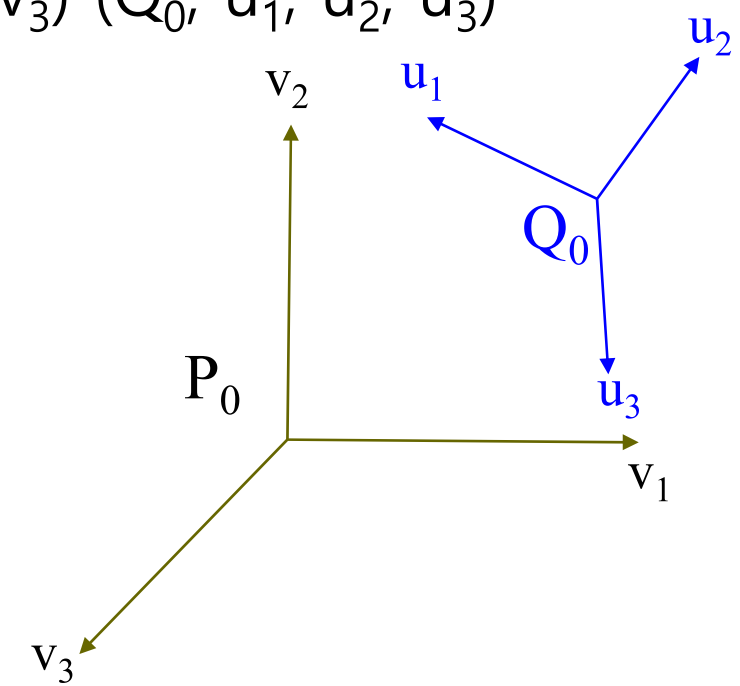
Change of Frames

- Consider two frames (P_0, v_1, v_2, v_3) (Q_0, u_1, u_2, u_3)

$$u_1 = \gamma_{11}v_1 + \gamma_{12}v_2 + \gamma_{13}v_3$$
$$u_2 = \gamma_{21}v_1 + \gamma_{22}v_2 + \gamma_{23}v_3$$
$$u_3 = \gamma_{31}v_1 + \gamma_{32}v_2 + \gamma_{33}v_3$$
$$Q_0 = \gamma_{41}v_1 + \gamma_{42}v_2 + \gamma_{43}v_3 + P_0$$

$$\begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix} = \mathbf{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

$$\mathbf{M} = \begin{bmatrix} \gamma_{11} & \gamma_{12} & \gamma_{13} & 0 \\ \gamma_{21} & \gamma_{22} & \gamma_{23} & 0 \\ \gamma_{31} & \gamma_{32} & \gamma_{33} & 0 \\ \gamma_{41} & \gamma_{42} & \gamma_{43} & 1 \end{bmatrix}$$



Change of Frames

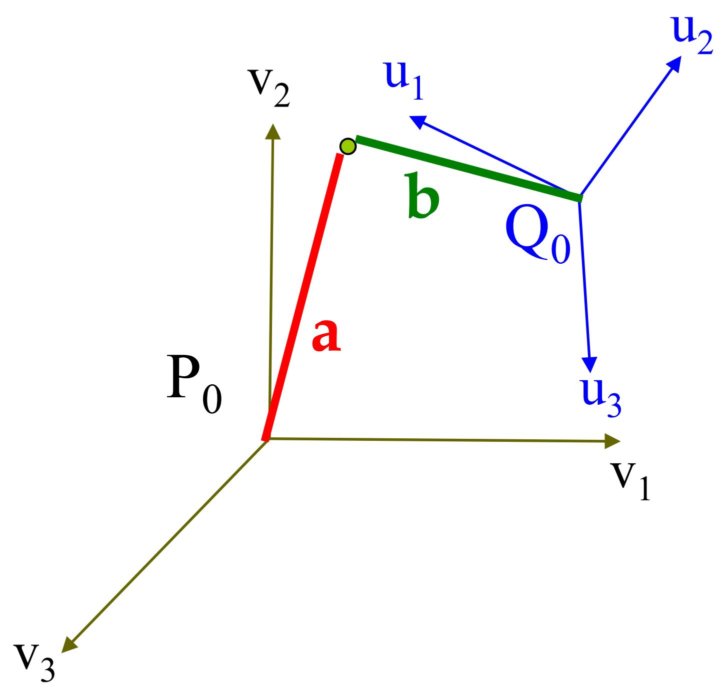
- Within the two frames (P_0, v_1, v_2, v_3) (Q_0, u_1, u_2, u_3) any point and vector has a representation of the same form

$$b^T \begin{bmatrix} u_1 \\ u_2 \\ u_3 \\ Q_0 \end{bmatrix} = b^T \mathrm{M} \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix} = a^T \begin{bmatrix} v_1 \\ v_2 \\ v_3 \\ P_0 \end{bmatrix}$$

$$\mathrm{M}^T = \begin{bmatrix} \gamma_{11} & \gamma_{21} & \gamma_{31} & \gamma_{41} \\ \gamma_{12} & \gamma_{22} & \gamma_{32} & \gamma_{42} \\ \gamma_{13} & \gamma_{23} & \gamma_{33} & \gamma_{43} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\therefore a = \mathrm{M}^T b$$

$$\therefore b = (\mathrm{M}^T)^{-1} a$$



OpenGL Frames

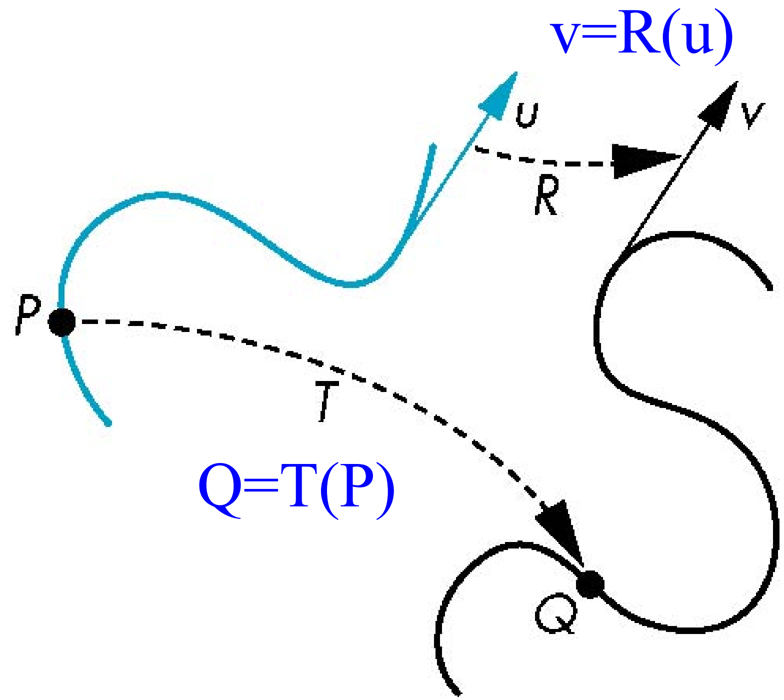
- Model-view coordinate system
- World coordinate system
- Camera coordinate system
- Clipping coordinate system
- Normalized device coordinate system
- Screen coordinate system

General Transformations

- A transformation maps points to other points and/or vectors to other vectors

$q = f(p)$

$v = f(u)$



Affine Transformations

- The affine transformation maintains collinearity.
  - That is, every affine transformation preserves lines. All points on a line exist on the transformed line.
- Also, it maintains the ratio of distance.
  - That is, the midpoint of a line is located at the midpoint of the transformed line segment.
- $P' = f(P)$
- $P' = f(\alpha P_1 + \beta P_2) = \alpha\, f(P_1) + \beta\, f(P_2)$

Affine Transformation

- Most transformation in computer graphics are affine transformation. Affine transformation include **translation, rotation, scaling, shearing**.
- The transformed point P′ (x′, y′, z′) can be expressed as a linear combination of the original point P (x, y, z), i.e.

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} & \alpha_{14} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} & \alpha_{24} \\ \alpha_{31} & \alpha_{32} & \alpha_{33} & \alpha_{34} \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

Affine Transformation

- The transformed point P′ (x′, y′, z′) can be expressed as a linear combination of the original point P (x, y, z), i.e.,

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{pmatrix} \alpha_{11} x + \alpha_{12} y + \alpha_{13} \\ \alpha_{21} x + \alpha_{22} y + \alpha_{23} \\ 1 \end{pmatrix}$$

↓

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{pmatrix} \alpha_{11} & \alpha_{12} & \alpha_{13} \\ \alpha_{21} & \alpha_{22} & \alpha_{23} \\ 0 & 0 & 1 \end{pmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Geometric Transformation

- Geometric transformation refers to a function that transforms a group of points describing a geometric object to new points.
- At this time, the points are transformed to a new position while maintaining the relationship between the vertices of the objects.
- Basic transformation
  - Translation
  - Rotation
  - Scaling

OpenGL Column-Major Order

- **2D transformation matrix, M**

$$\mathbf{M} = \begin{bmatrix} a & b \\ c & d \end{bmatrix}$$

- **If Point p is a column vector (OpenGL) :**

$$\mathbf{p}' = \mathbf{M}\mathbf{p}$$

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

- If Point p is a row vector:

$$\mathbf{p}' = \mathbf{p}\mathbf{M}^{\mathrm{T}}$$

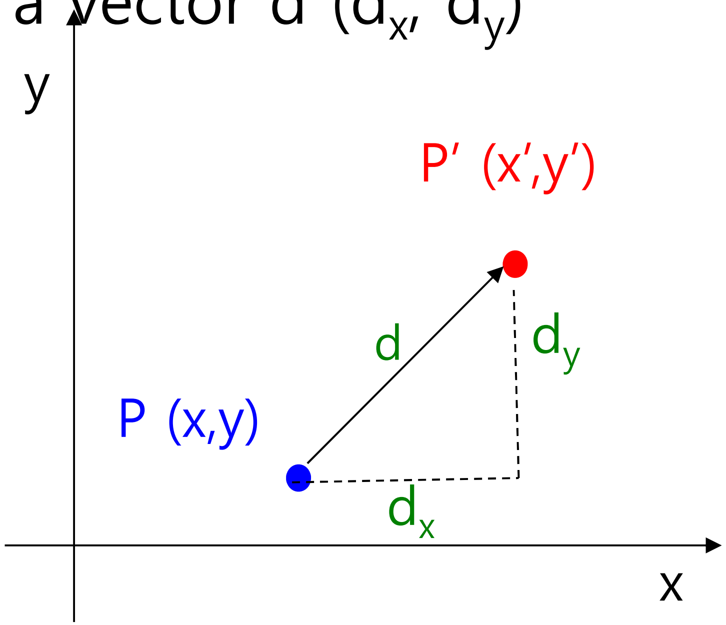
$$\begin{bmatrix} x' & y' \end{bmatrix} = \begin{bmatrix} x & y \end{bmatrix} \begin{bmatrix} a & c \\ b & d \end{bmatrix}$$

2D Translation

- Translation moves a point P(x, y) to a new location P′(x′, y′)
- Displacement determined by a vector d (d_x, d_y)

  $x' = x + d_x$

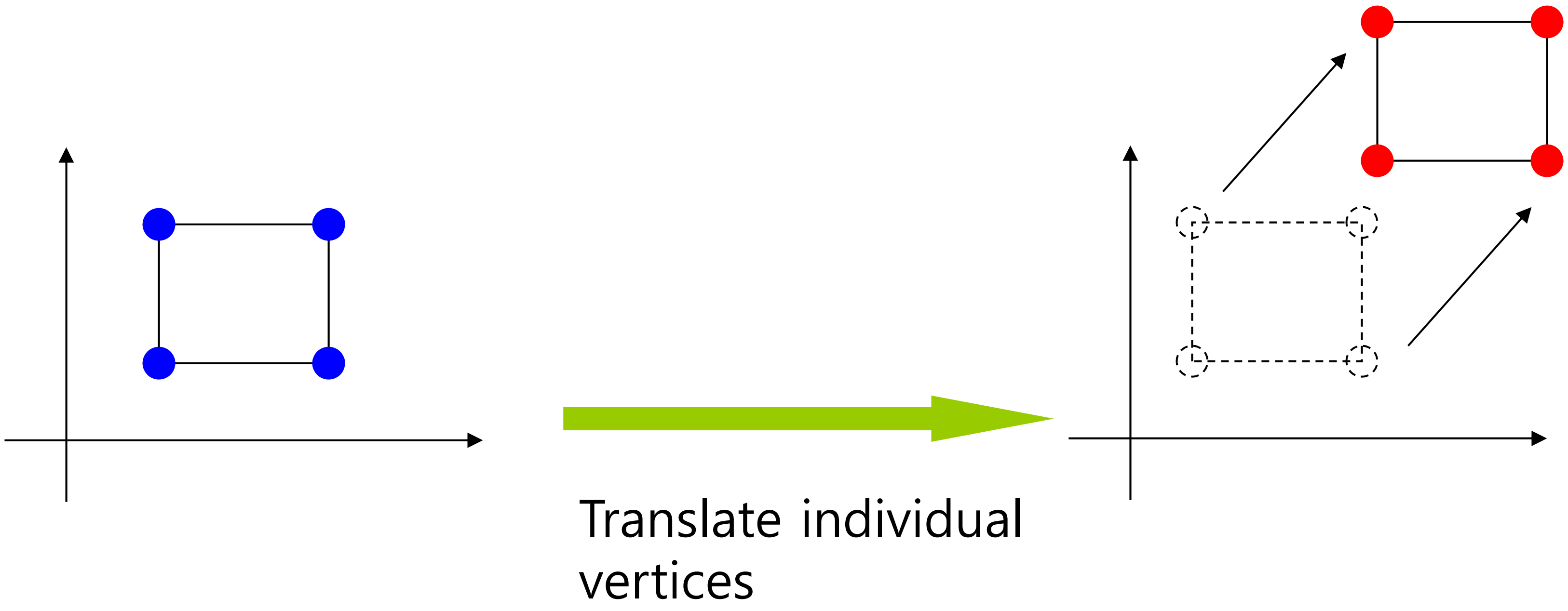
  $y' = y + d_y$



$$P' = P + d \quad where \quad P' = \begin{pmatrix} x' \\ y' \end{pmatrix} P = \begin{pmatrix} x \\ y \end{pmatrix} d = \begin{pmatrix} d_x \\ d_y \end{pmatrix}$$

2D Translation

- What if you move an object with multiple vertices?



2D Translation

- Using the homogeneous coordinate representation in some frame

  $\mathbf{p}=[\ x\ y\ 1]^T$

  $\mathbf{p}'=[x'\ y'\ 1]^T$

  $\mathbf{d}=[dx\ dy\ 0]^T$

- Hence $\mathbf{p}' = \mathbf{p} + \mathbf{d}$ or

  $x'=x+d_x$

  $y'=y+d_y$

Note that this expression is in four dimensions and expresses point = vector + point

2D Translation

- We can also express 2D translation using a 3 x 3 matrix **T** in homogeneous coordinates:

$\mathbf{p}' = \mathbf{T}\mathbf{p}$ where

$$\mathbf{T} = \mathbf{T}(d_x, d_y) = \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix}$$

- This form is better for implementation because all affine transformations can be expressed this way and multiple transformations can be concatenated together.

2D Translation

- **2D translation**

$$x' = x + d_x$$
$$y' = y + d_y$$

- **Inverse translation**

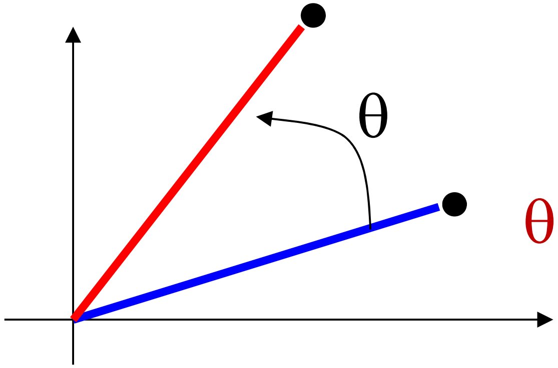
$$x = x' - d_x$$
$$y = y' - d_y$$

- **Identity translation**

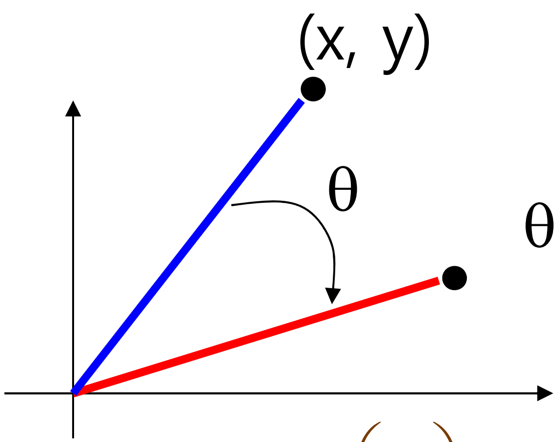
$$x' = x + 0$$
$$y' = y + 0$$

Rotation

- 2D rotation about the origin by θ



θ > 0 : Rotate counter clockwise in RHS



θ < 0 : Rotate clockwise in LHS

$$\theta = \arctan\left(\frac{y}{x}\right)$$

2D Rotation

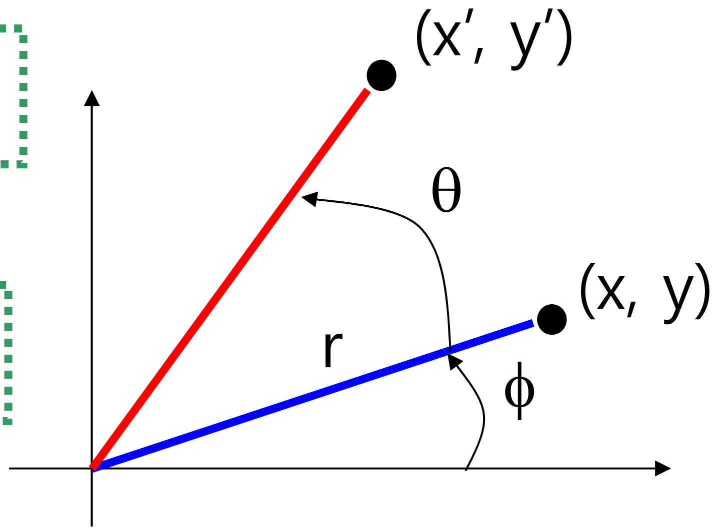
- Rotation of a point P(x,y) by θ about an origin (0,0)

$x = r\cos(\phi)$ $\quad y = r\sin(\phi)$

$x' = r\cos(\phi + \theta)$ $\quad y' = r\sin(\phi + \theta)$

$$
\begin{aligned}
x' &= r\cos(\phi + \theta) \\
&= r\cos(\phi)\cos(\theta) - r\sin(\phi)\sin(\theta) \\
&= x\cos(\theta) - y\sin(\theta)
\end{aligned}
$$

$$
\begin{aligned}
y' &= r\sin(\phi + \quad) \\
&= r\sin(\phi)\cos(\theta) + r\cos(\phi)\sin(\theta) \\
&= y\cos(\theta) + x\sin(\theta)
\end{aligned}
$$



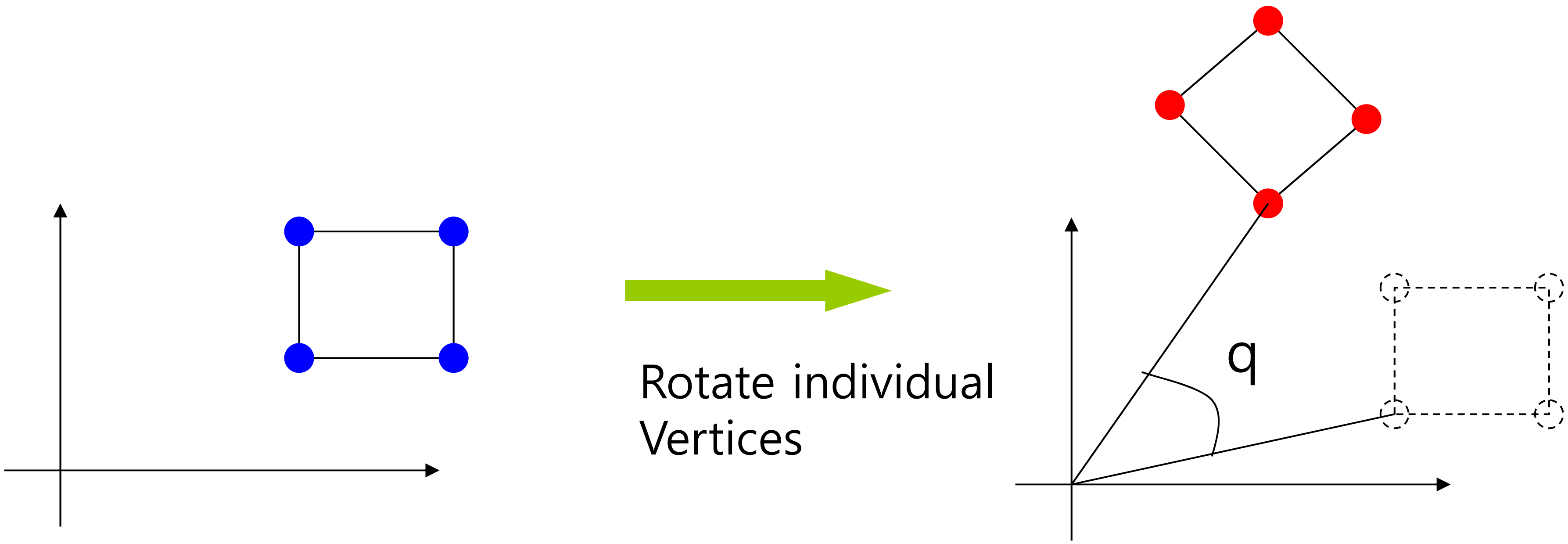
$x' = x\cos(\theta) - y\sin(\theta)$

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

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

2D Rotation

- What if you rotate an object with multiple vertices?



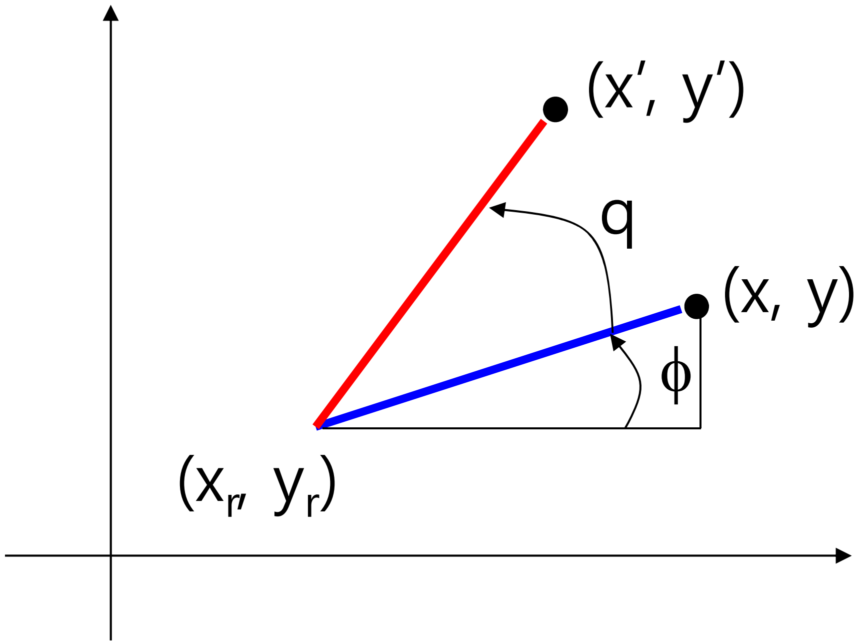
2D Rotation about an Arbitrary Pivot

- Rotation of a point P(x,y) by θ about an arbitrary pivot point, (x_r, y_r) :

  $P' = R(\theta)\ P$

  $x' = x_r + (x - x_r) \cos\theta - (y - y_r) \sin\theta$

  $y' = y_r + (x - x_r) \sin\theta + (y - y_r) \cos\theta$



2D Rotation

- **2D rotation**

$$R = \begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix}$$

- **Inverse rotation**

$$R^{-1} = \begin{pmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{pmatrix}$$

- **Identity rotation**

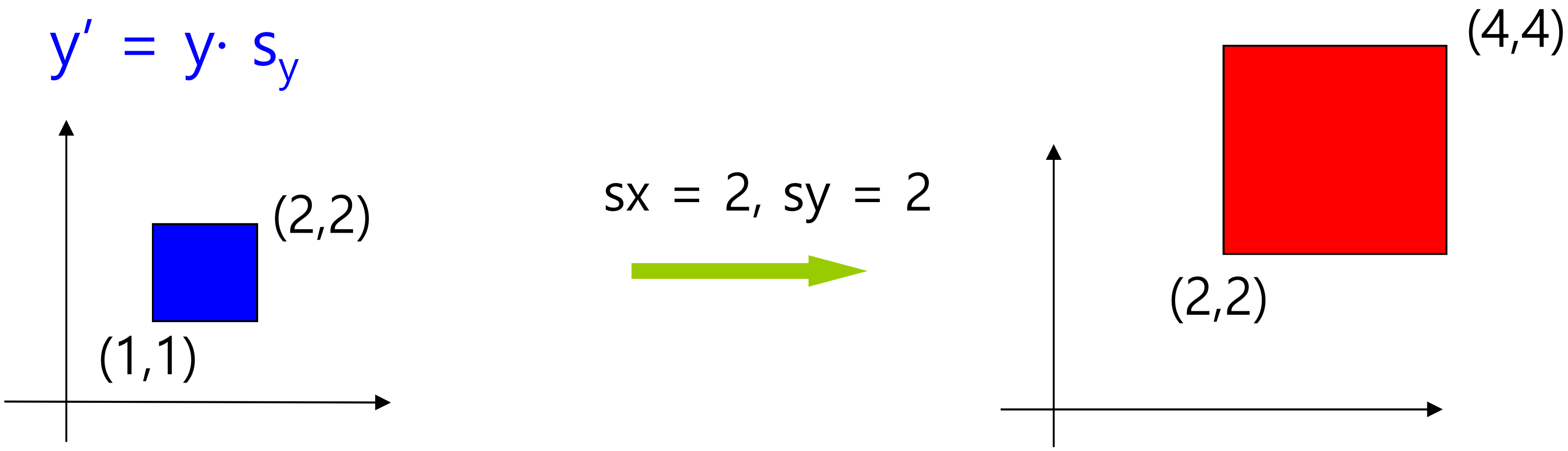
$$R_{\theta=0} = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

2D Scale

- Scaling makes an object larger or smaller by a scaling factor (s_x, s_y). This is affine non-rigid-body transformation. Scaling by 1 does not change an object.
- Scaling is done by an origin. Scaling changes not only the size of object, but also the position of object.

$x' = x \cdot s_x$

$y' = y \cdot s_y$



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

2D Scale about an Arbitrary Pivot

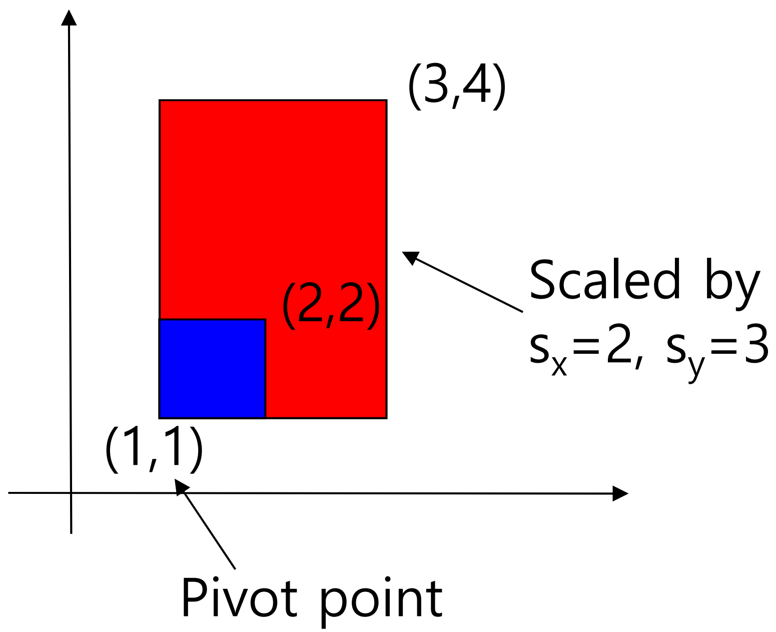
- Scale a point P(x,y) by a scaling factor relative to an arbitrary pivot point, (x_f, y_f) : $P' = S(s_x, s_y)\ P$

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

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

$$x' = x\ s_x + x_f\ (1 - s_x)$$

$$y' = y\ s_y + y_f\ (1 - s_y)$$



2D Scale

- **2D scale**

$$S = \begin{pmatrix} s_x & 0 \\ 0 & s_y \end{pmatrix}$$

- **Inverse scale**

$$S^{-1} = \begin{pmatrix} 1/s_x & 0 \\ 0 & 1/s_y \end{pmatrix}$$

- **Identity scale**

$$S = \begin{pmatrix} 1 & 0 \\ 0 & 1 \end{pmatrix}$$

2D Reflection (Mirror)

- Reflection is the transformation of an object in opposite direction with respect to a fixed point.
  - Reflection preserves angles and lengths.
- 2D reflection over x axis

  $x' = x$

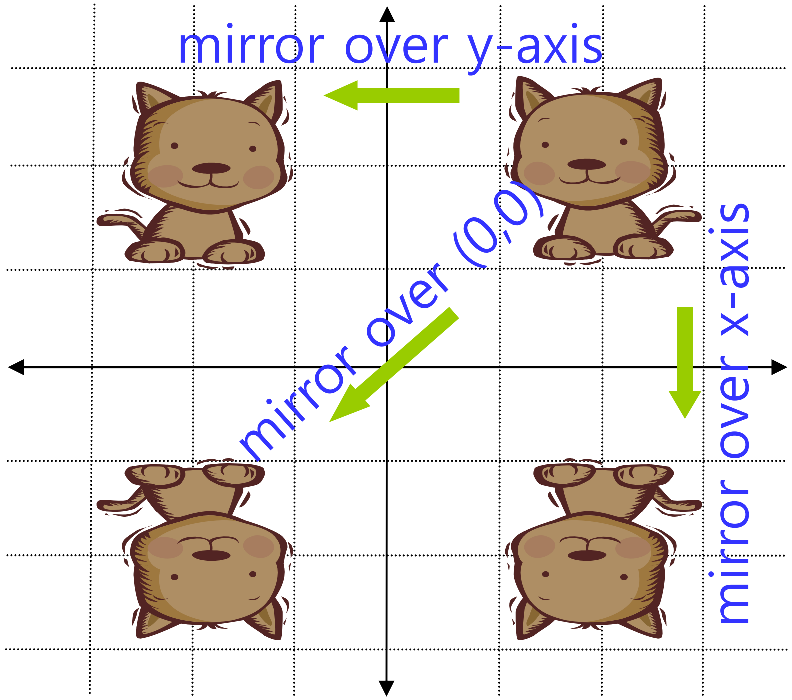
  $y' = -y$
- 2D reflection over y axis

  $x' = -x$

  $y' = y$
- 2D reflection over (0,0)

  $x' = -x$

  $y' = -y$



2D Reflection (Mirror)

- 2D reflection over a line, $y = x$

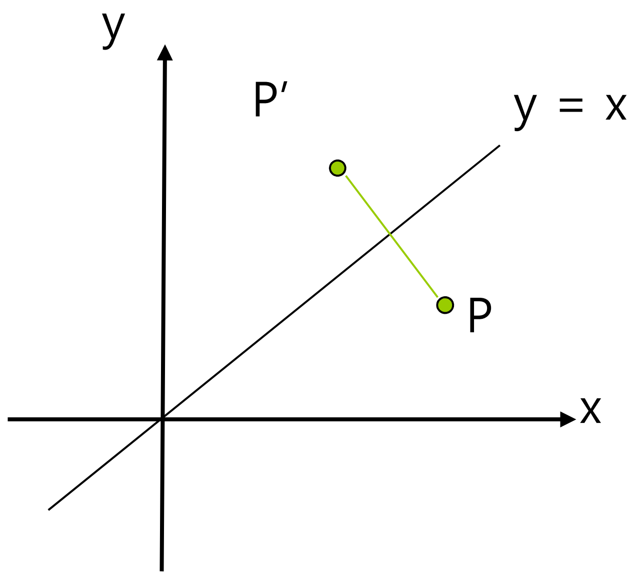
  $x' = y$

  $y' = x$

- 2D reflection over a line, $y = -x$

  $x' = -y$

  $y' = -x$



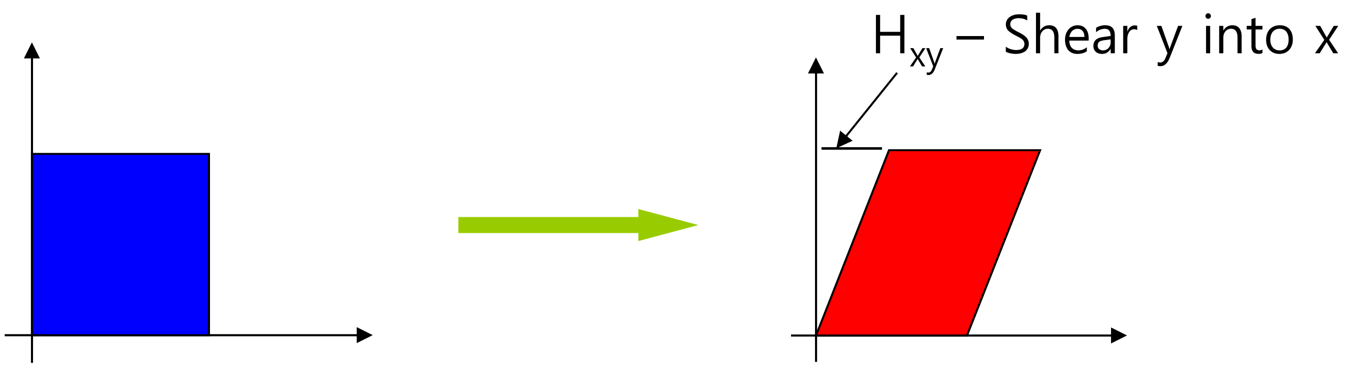
2D Shearing

- The Y-axis is not changed, and shearing applied in the X-axis direction:

  $x' = x + y \cdot h_{xy}$

  $y' = y$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & h_{xy} & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



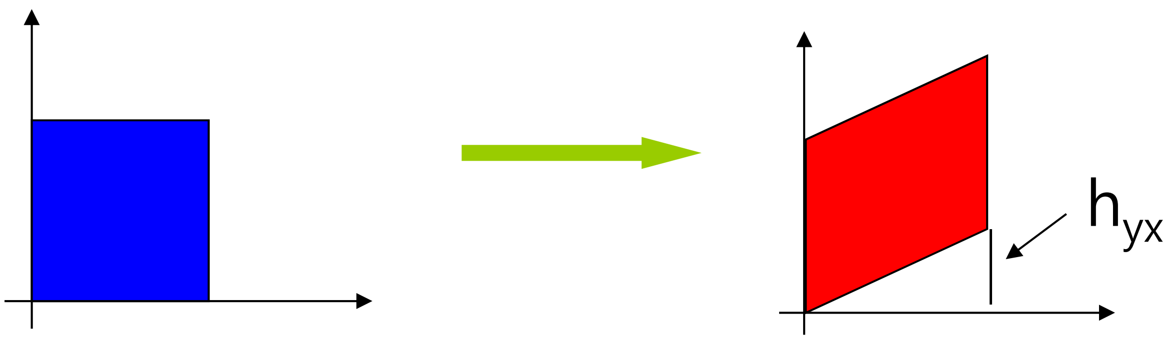
2D Shearing

- Shearing transformation does not change the size of object.
- The X-axis is not changed, and shearing applied in the Y-axis direction :

$x' = x$

$y' = x \cdot h_{yx} + y$

$$\begin{pmatrix} x' \\ y' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 \\ h_{yx} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$



Homogeneous Coordinates

- In order to multiply translation, rotation, scaling transformation matrix, homogeneous coordinates are used.
- In homogeneous coordinates, the two-dimensional point P (x, y) is expressed as P(x, y, w).
- (1, 2, 3) and (2, 4, 6) represent the same homogeneous coordinates.
- If the w of the point P (x, y, w) is 0, the point is located at an infinite point. If w is not 0, the point can be expressed as (x/w, y/w, 1).

Transforming Homogeneous Coordinates

$$T(dx, dy) = \begin{pmatrix} 1 & 0 & dx \\ 0 & 1 & dy \\ 0 & 0 & 1 \end{pmatrix}$$

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

$$S(sx, sy) = \begin{pmatrix} sx & 0 & 0 \\ 0 & sy & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

- The two-dimensional transformation matrix can be expressed as a 3x3 matrix of homogeneous coordinates.

3x3 2D Translation Matrix

- Matrix-vector multiplication

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} d_x \\ d_y \end{bmatrix}$$

↓

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

3x3 2D Rotation Matrix

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

↓

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

3x3 2D Scale Matrix

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

↓

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

3x3 2D Shearing Matrix

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{pmatrix} 1 & h_{xy} \\ h_{yx} & 1 \end{pmatrix} \begin{bmatrix} x \\ y \end{bmatrix}$$

$$\Downarrow$$

$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{pmatrix} 1 & h_{xy} & 0 \\ h_{yx} & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} * \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

Inverse 2D Transformation Matrix

$$T^{-1} = \begin{pmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{pmatrix}$$

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

$$S^{-1} = \begin{pmatrix} 1/s_x & 0 & 0 \\ 0 & 1/s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

Composing Transformation

- *Composing transformation* is a process of forming one transformation by applying several transformation in sequence.
- If you want to transform one point, apply one transformation at a time or multiply the matrix and then multiply this matrix by the point.

$$Q = (M3 \cdot (M2 \cdot (M1 \cdot P))) = \underbrace{M3 \cdot M2 \cdot M1}_{M} \cdot P$$

(pre-multiply)

- Matrix multiplication is associative.

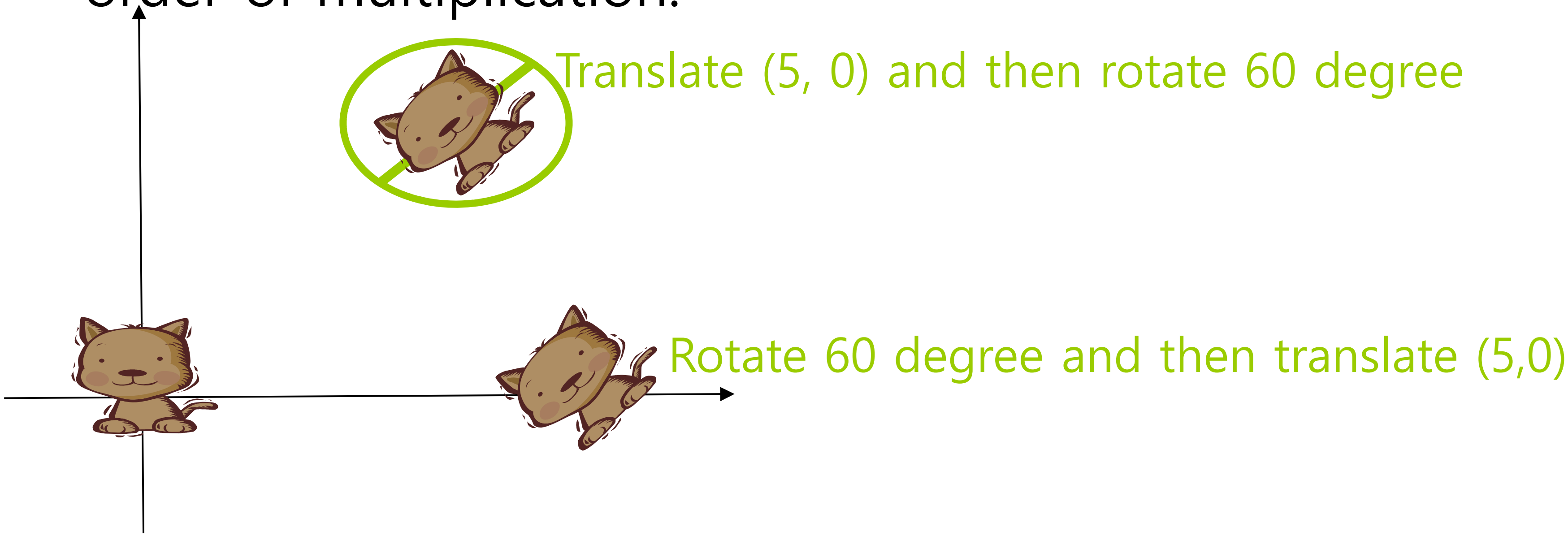
$$M3 \cdot M2 \cdot M1 = (M3 \cdot M2) \cdot M1 = M3 \cdot (M2 \cdot M1)$$

- Matrix multiplication is not commutative.

$$A \cdot B \;\; != \;\; B \cdot A$$

Transformation Order Matters!

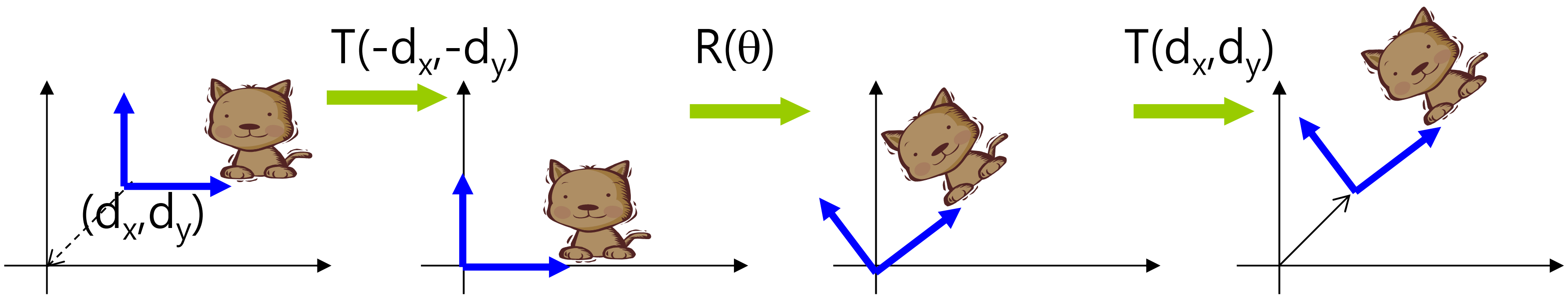
- The multiplication of the transformation matrix is not commutative.
- Even if the transformation matrix is the same, it may have completely different results depending on the order of multiplication.



2D Rotate about an Arbitrary Pivot

- Two-dimensional rotation by θ at an arbitrary pivot point $P(d_x, d_y)$:
  1. $T(-d_x, -d_y)$
  2. $R(\theta)$
  3. $T(d_x, d_y)$

$$\begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} \cos\theta & -\sin\theta & d_x(1-\cos\theta)+d_y\sin\theta \\ \sin\theta & \cos\theta & d_y(1-\cos\theta)-d_x\sin\theta \\ 0 & 0 & 1 \end{pmatrix}$$

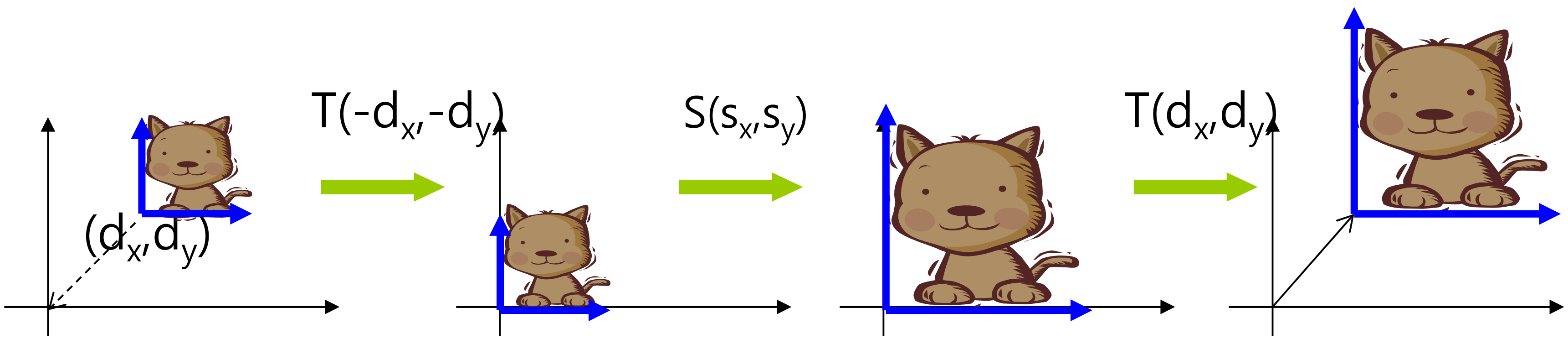


$T(-d_x,-d_y)\cdot R(\theta)\cdot T(d_x,d_y) = R(d_x, d_y, \theta)$

2D Scale about an Arbitrary Pivot

- Two-dimensional scaling an arbitrary pivot point $P(d_x, d_y)$:
  1. $T(-d_x, -d_y)$
  2. $S(s_x, s_y)$
  3. $T(d_x, d_y)$

$$\begin{pmatrix} 1 & 0 & d_x \\ 0 & 1 & d_y \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & -d_x \\ 0 & 1 & -d_y \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x & 0 & d_x(1 - s_x) \\ 0 & s_y & d_y(1 - s_y) \\ 0 & 0 & 1 \end{pmatrix}$$

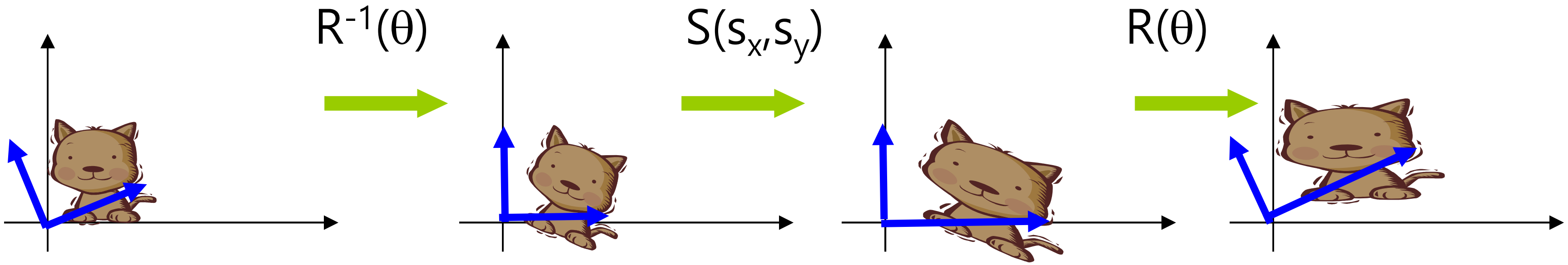


$T(-d_x,-d_y)\cdot S(s_x, s_y)\cdot T(d_x,d_y) = S(d_x, d_y, s_x, s_y)$

2D Scale in an Arbitrary Direction

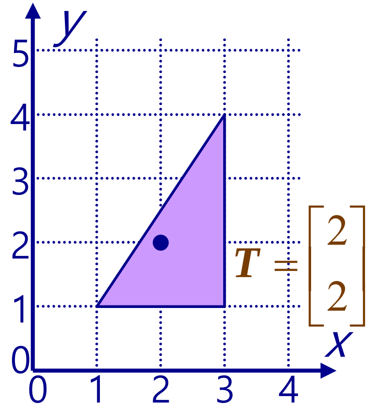
- Two dimensional scaling in an arbitrary direction (Rotating *the object to align the desired scaling directions with the coordinate axes* before scale transformation)
  1. $R^{-1}(\theta)$
  2. $S(s_x, s_y)$
  3. $R(\theta)$

$$\begin{pmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} s_x & 0 & 0 \\ 0 & s_y & 0 \\ 0 & 0 & 1 \end{pmatrix}\begin{pmatrix} \cos\theta & \sin\theta & 0 \\ -\sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} = \begin{pmatrix} s_x\cos^2\theta + s_y\sin^2\theta & (s_x - s_y)\cos\theta\sin\theta & 0 \\ (s_x - s_y)\cos\theta\sin\theta & s_y\cos^2\theta + s_x\sin^2\theta & 0 \\ 0 & 0 & 1 \end{pmatrix}$$



Example: 2D Rotate about an Arbitrary Pivot

Rotate a triangle with vertices (1,1), (3,1), (3,4) by 45 degrees about the pivot point (2,2).



1. Translate point to origin $\mathbf{T}_1 = \begin{bmatrix} -2 \\ -2 \end{bmatrix}$
2. Rotate 45 degrees $\mathbf{R} = \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix}$
3. Translate back to original location $\mathbf{T}_2 = \begin{bmatrix} 2 \\ 2 \end{bmatrix}$
4. Composite transformation $P' = R(P + T_1) + T_2$

$$P' = \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \left(\begin{bmatrix} x \\ y \end{bmatrix} + \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right) + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

Example: 2D Rotate about an Arbitrary Pivot

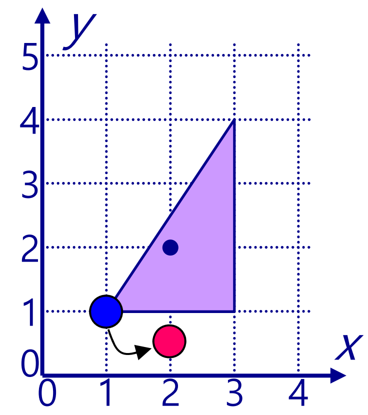
- P_1 (1, 1)

$$P_1' = \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \left(\begin{bmatrix} 1 \\ 1 \end{bmatrix} + \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right) + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \left(\begin{bmatrix} -1 \\ -1 \end{bmatrix} \right) + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 0 \\ -1.414 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 2 \\ 0.586 \end{bmatrix}$$



Example: 2D Rotate about an Arbitrary Pivot

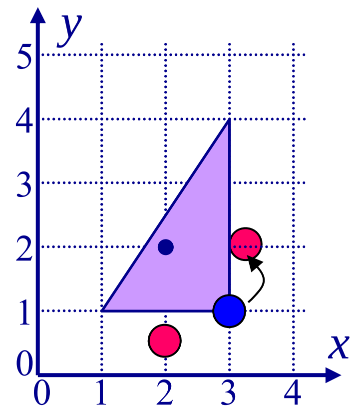
- P_2 (3, 1)

$$P_2' = \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \left(\begin{bmatrix} 3 \\ 1 \end{bmatrix} + \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right) + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \left(\begin{bmatrix} 1 \\ -1 \end{bmatrix} \right) + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 1.414 \\ 0 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} 3.414 \\ 2 \end{bmatrix}$$



Example: 2D Rotate about an Arbitrary Pivot

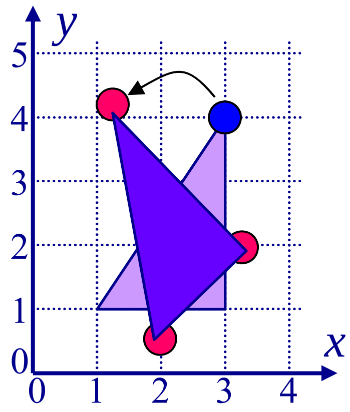
- P_3 (3, 4)

$$P_3' = \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \left(\begin{bmatrix} 3 \\ 4 \end{bmatrix} + \begin{bmatrix} -2 \\ -2 \end{bmatrix} \right) + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

$$= \begin{bmatrix} .707 & -.707 \\ .707 & .707 \end{bmatrix} \left(\begin{bmatrix} 1 \\ 2 \end{bmatrix} \right) + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

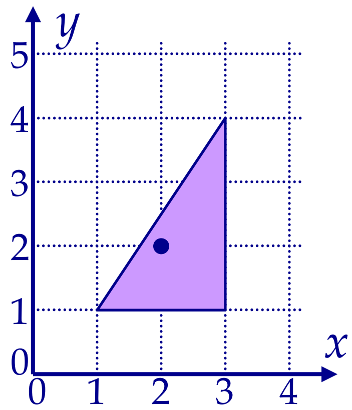
$$= \begin{bmatrix} -.707 \\ 2.121 \end{bmatrix} + \begin{bmatrix} 2 \\ 2 \end{bmatrix}$$

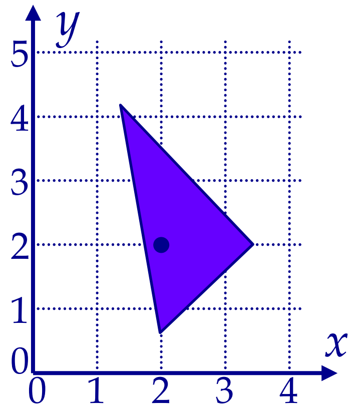
$$= \begin{bmatrix} 1.293 \\ 4.121 \end{bmatrix}$$



Example: 2D Rotate about an Arbitrary Pivot

- Result:





Before:
(1, 1), (3, 1), (3, 4)

After:
(2, 0.59), (3.41, 2), (1.29, 4.2)

Example: 2D Rotate about an Arbitrary Pivot Using Composite Transformation Matrix

- Rotate a triangle with vertices (1,1), (3,1), (3,4) by 45 degrees about the pivot point (2,2).
- P′= T(2,2)R(45)T(-2,-2)P = M P

$$M = T_{(2,2)} R_{45} T_{(-2,-2)})$$

$$= \left(\begin{bmatrix} 1 & 0 & 2 \\ 0 & 1 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos(\mathbf{4}5^\circ) & -\sin(\mathbf{4}5^\circ) & 0 \\ \sin(\mathbf{4}5^\circ) & \cos(\mathbf{4}5^\circ) & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix} \right)$$

$$= \begin{bmatrix} .707 & -.707 & 2 \\ .707 & .707 & 2 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -2 \\ 0 & 1 & -2 \\ 0 & 0 & 1 \end{bmatrix}$$

$$= \begin{bmatrix} .707 & -.707 & 2 \\ .707 & .707 & -.828 \\ 0 & 0 & 1 \end{bmatrix} = \mathbf{M}$$

Example: 2D Rotate about an Arbitrary Pivot Using Composite Transformation Matrix

1. P_1

$$P_1' = MP_1 = \begin{bmatrix} .707 & -.707 & 2 \\ .707 & .707 & -.828 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 2 \\ .586 \\ 1 \end{bmatrix}$$

2. P_2

$$P_2' = MP_2 = \begin{bmatrix} .707 & -.707 & 2 \\ .707 & .707 & -.828 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 1 \\ 1 \end{bmatrix} = \begin{bmatrix} 3.414 \\ 2 \\ 1 \end{bmatrix}$$

3. P_3

$$P_3' = MP_3 = \begin{bmatrix} .707 & -.707 & 2 \\ .707 & .707 & -.828 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 3 \\ 4 \\ 1 \end{bmatrix} = \begin{bmatrix} 1.293 \\ 4.121 \\ 1 \end{bmatrix}$$