이미지 변환

Image conversion

직선의 표현

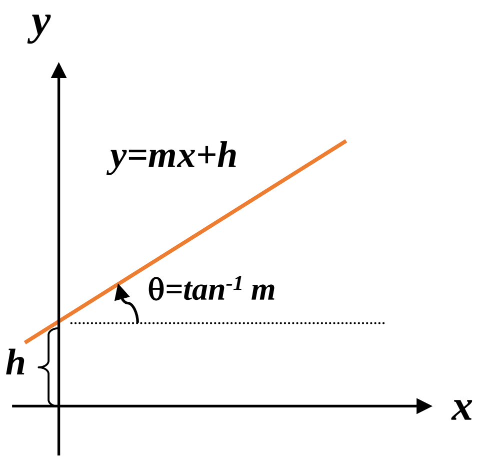
양함수 / 음함수

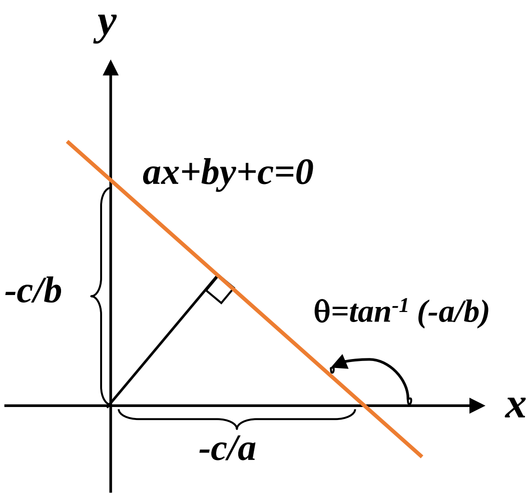
- 양함수 : 종속변수 없이 독립변수들의 식만으로 표현되는 함수

$$y = f(x)\,, \qquad y = mx + h$$

- 음함수: 종속변수가 독립변수와 분리되지 않은 하나의 관계식으로 주어진 함수

$$P(x, y) = 0\,, \qquad ax + by + c = 0$$





기울기(slope)

$$y = mx + h$$

y의 절편(intercept)

기울기가 무한대인 수직선은 표현 불가능

$$x = \frac{1}{m}(y - h)$$

수평선 표현 불가능

$$ax + by + c = 0$$

$$\begin{cases} a = 0 : \text{수직선} \\ b = 0 : \text{수평선} \end{cases}$$

행렬 곱 계산의 예

행렬(matrix)

- 직사각형 모양으로 구성된 수의 배열

  > 행(row): 행렬의 수평 부분 배열
  > 열(column): 행렬의 수직 부분 배열

행렬 곱의 예 1

$$A=\begin{bmatrix}2 & 1 & 3\end{bmatrix} \quad B=\begin{bmatrix}1 & 0 & 2\\ -2 & 3 & 1\\ 4 & 6 & 3\end{bmatrix}$$

2×1+1 ×(-2)+3 × 4=12

2×2+1 ×1+3 × 3=14

$$A\times B=\begin{bmatrix}2 & 1 & 3\end{bmatrix}\begin{bmatrix}1 & 0 & 2\\ -2 & 3 & 1\\ 4 & 6 & 3\end{bmatrix}=\begin{bmatrix}12 & 21 & 14\end{bmatrix}$$

2×0+1 ×3+3 × 6=21

행렬 곱의 예 2

$$A=\begin{bmatrix}0 & 2 & -1\\ 3 & 1 & 2\\ 1 & 2 & 1\end{bmatrix} \quad B=\begin{bmatrix}1 & 0 & 2\\ -2 & 3 & 1\\ 4 & 6 & 3\end{bmatrix}$$

0×2+2×1+(-1)×3=-1

0×0+2×3+(-1)×6=0

0×1+2×(-2)+(-1)×4=-8

$$A\times B=\begin{bmatrix}0 & 2 & -1\\ 3 & 1 & 2\\ 1 & 2 & 1\end{bmatrix}\begin{bmatrix}1 & 0 & 2\\ -2 & 3 & 1\\ 4 & 6 & 3\end{bmatrix}=\begin{bmatrix}-8 & 0 & -1\\ 9 & 15 & 13\\ 1 & 12 & 7\end{bmatrix}$$

3×1+1×(-2)+2×4=9

행렬 표현

행렬 표현(matrix representation)

- 조합 변환(composite transformation)이 일어나는 경우 행렬의 순서에 주의

  > 영어를 읽는 순서의 반대로 계산 (우축→좌축 순서로 연산 적용)
  > 행렬 변환 연산을 사용하여 행 벡터에 의해 열 벡터로 대체
  > 행렬 곱은 교환 법칙이 성립되지 않음

예

$$C \quad B \quad A$$

❶ ❷

$$A = T(-x_f, -y_f)$$

$$B = R(\theta)$$

$$C = T(x_f, y_f)$$

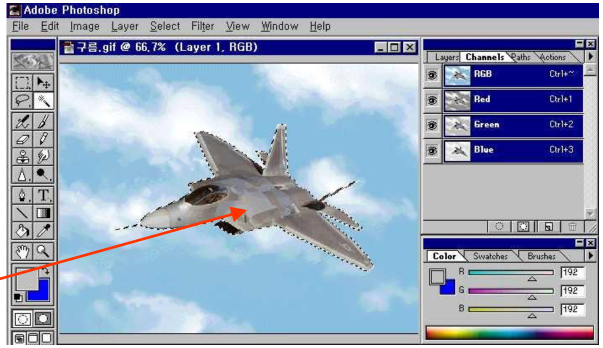
❷ ❶

$$M = CBA = \begin{bmatrix} 1 & 0 & x_f \\ 0 & 1 & y_f \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} \cos\theta & -\sin\theta & 0 \\ \sin\theta & \cos\theta & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -x_f \\ 0 & 1 & -y_f \\ 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta & y_f \sin\theta - x_f \cos\theta + x_f \\ \sin\theta & \cos\theta & -x_f \sin\theta - y_f \cos\theta + y_f \\ 0 & 0 & 1 \end{bmatrix}$$

이미지 변환의 예

PhotoShop에서의 작업 예

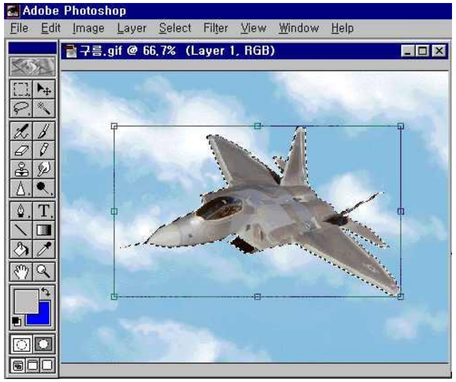
이미지 선택



File>Open
이미지 선택

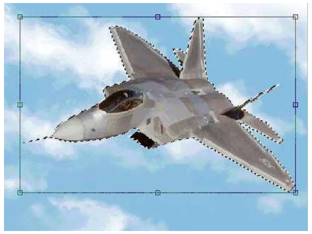
변형할 이미지를
선택한다.

조절점 선택

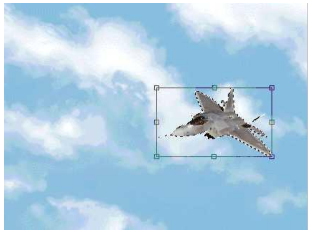


Edit>Free Transform
8개의 **조절점** 선택

확대/축소 하기

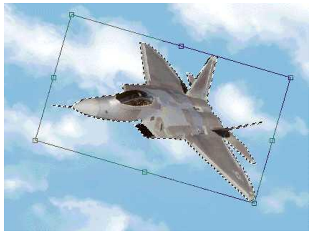


마우스 포인터를 조절점 위로 가져 가면
크기를 조정하는 화살표로 바뀐다.
이때 드래그하여 확대/축소 시킨다.
Shift 키를 누른 상태에서는
가로, 세로 비율이 같게 되며,
작업 취소는 Esc 키를 누른다.



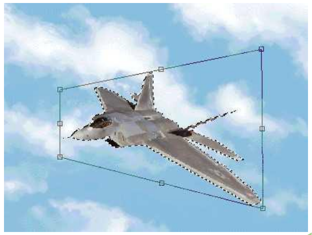
원하는 결과물을 얻으면 선택된
영역의 중앙을 더블 클릭한다.

회전(rotate) 하기



마우스 포인터를 조절점 바깥쪽으로
가져 가면 회전 각도를 조정하는 화살표로
바뀐다. 이때 드래그하여 회전시킨다.
Shift 키를 누른 상태에서는
0, 45, 90, 135, 180도 단위로 회전이 가능

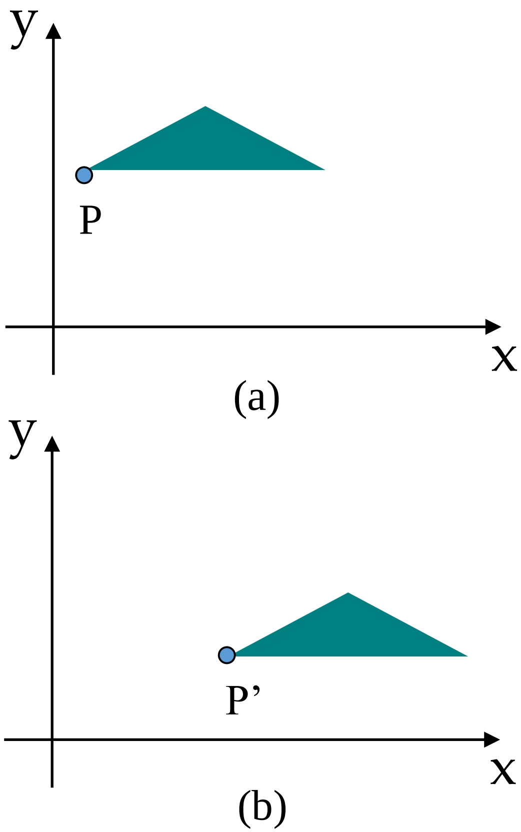
왜곡(distort) 하기



Ctrl 키를 누른 상태에서 마우스 포인터를
조절점 위로 가져 가면 형태를 조정하는
화살표로 바뀐다.
이때 드래그하여 변경시킨다.

2차원 변환

이동 변환 (Translation)

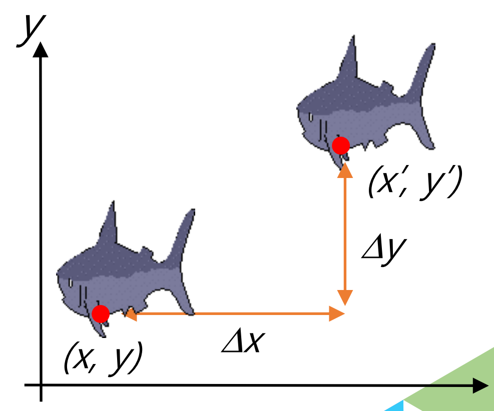


$$x' = x + T_x,$$
$$y' = y + T_y$$

$$P = \begin{bmatrix} x \\ y \end{bmatrix},\ P' = \begin{bmatrix} x' \\ y' \end{bmatrix},\ T = \begin{bmatrix} T_x \\ T_y \end{bmatrix}$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

$$x' = x + \Delta x$$
$$y' = y + \Delta y$$

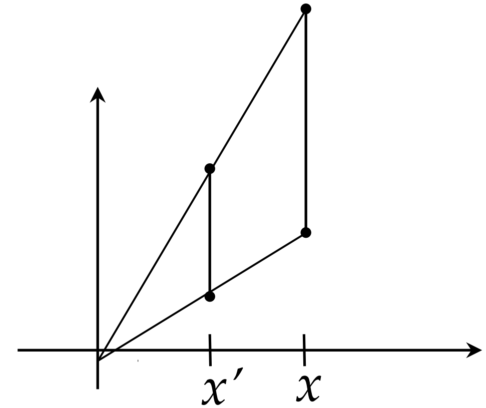


2차원 변환

크기 변환 (Scaling)

$$x' = x \cdot Sx\,,\qquad y' = y \cdot Sy$$

$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix} \cdot \begin{bmatrix} x \\ y \end{bmatrix}$$

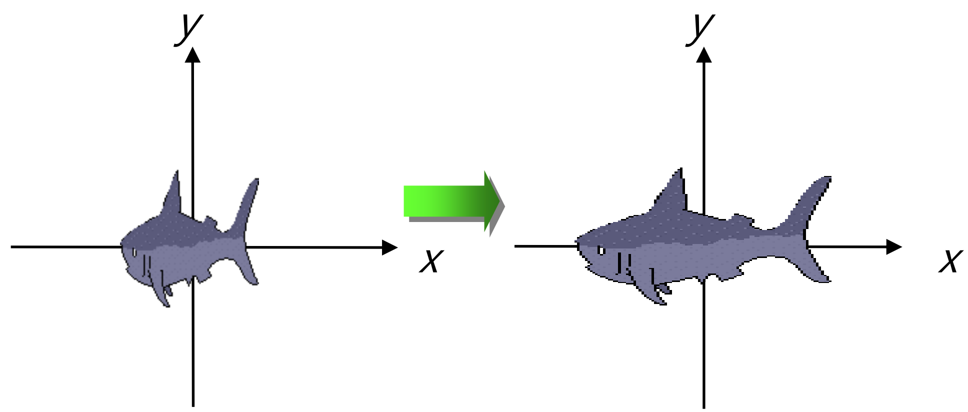


$$x' = \alpha x$$
$$y' = \beta y$$

크기 변환 상수(scaling constant) 크기에 대한...

- 크기 변형 상수 > 1 : 확장(expansion)
- 0 < 크기 변형 상수 < 1 : 축소(contraction)
- 크기 변형 상수 < 0 : 반사(reflection)

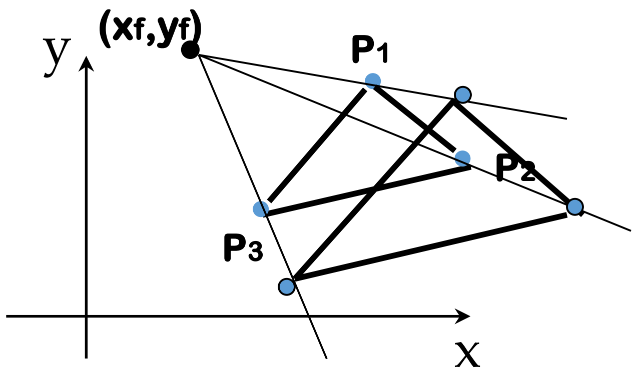
$$S = \begin{bmatrix} \alpha & 0 \\ 0 & \beta \end{bmatrix}$$



2차원 변환

크기 변환 (Scaling)

$$x' = x_f + (x - x_f)s_x\,, \qquad y' = y_f + (y - y_f)s_y$$



(x_f, y_f) : fixed point

$$P' = T(x_f, y_f) \cdot S(s_x, s_y) \cdot T(-x_f, -y_f) \cdot P$$

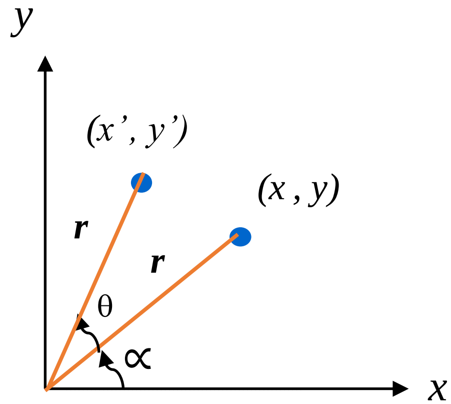
$$\begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 1 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} S_x & 0 & 0 \\ 0 & S_y & 0 \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} 1 & 0 & -T_x \\ 0 & 1 & -T_y \\ 0 & 0 & 1 \end{bmatrix} \cdot \begin{bmatrix} x \\ y \\ 1 \end{bmatrix}$$

2차원 변환

회전 변환 (Rotation)

$P' = R \cdot P$

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



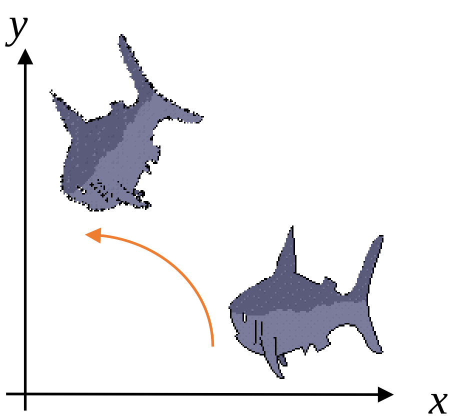
좌표중심을 회전점으로
각 θ 만큼 회전

$x = r\cos\propto,\ \ y = r\sin\propto$

$x' = r\cos(\propto + \theta) = r\cos\propto\cos\theta - r\sin\propto\sin\theta$

$y' = r\sin(\propto + \theta) = r\cos\propto\sin\theta + r\sin\propto\cos\theta$

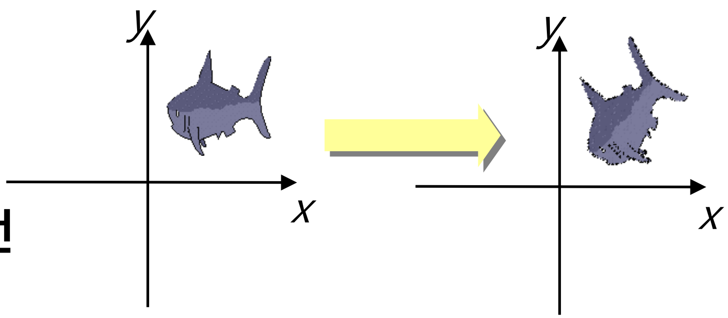
$\therefore\ \ x' = x\cos\theta - y\sin\theta,\ \ y' = x\sin\theta + y\cos\theta$

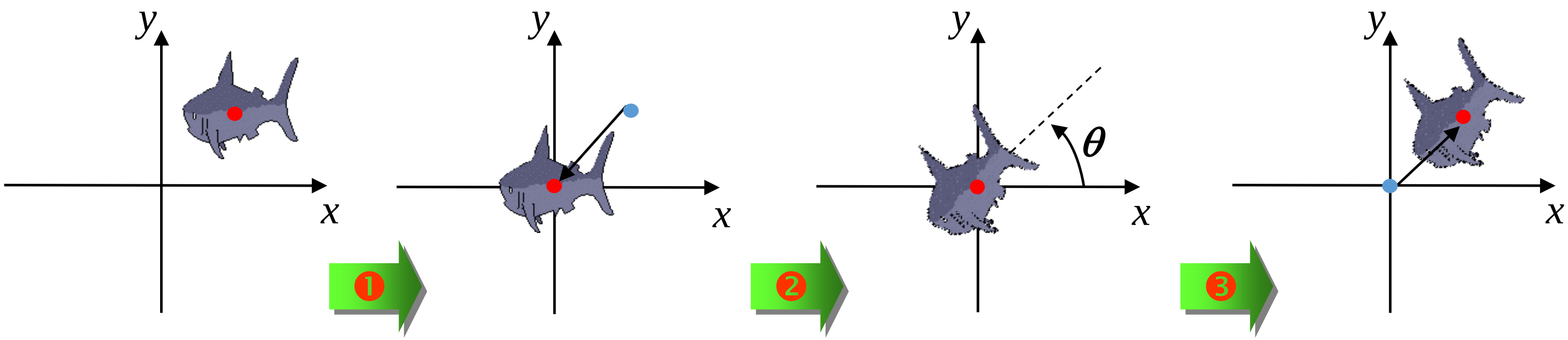


2차원 변환

고정점에 대한 회전

- 회전의 중심점이 원점이 아닌 경우
  1. 고정점을 원점으로 전환
  2. 전환된 점을 원점을 중심점으로 각 θ 만큼 회전
  3. 중심점을 원래의 고정점으로 이동





$$\bar{x} = x - x_f \qquad \bar{x} = \bar{x}\cos\theta - \bar{y}\sin\theta \qquad x' = \bar{x} + x_f$$
$$\bar{y} = y - y_f \qquad \bar{y} = \bar{x}\sin\theta + \bar{y}\cos\theta \qquad y' = \bar{y} + y_f$$

일반적인 회전 방정식

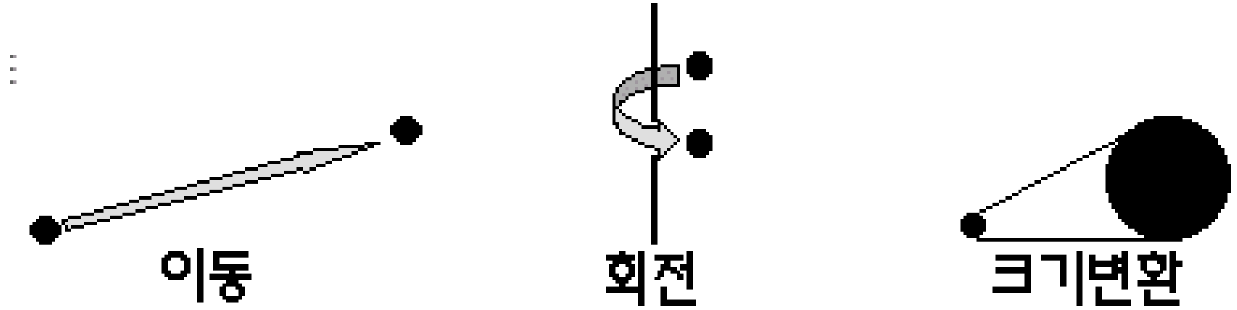
$$x' = (x - x_f)\cos\theta - (y - y_f)\sin\theta + x_f$$
$$y' = (x - x_f)\sin\theta + (y - y_f)\cos\theta + y_f$$

3차원 변환

- 기하학적 변환
  > 모델링하고 배치하고자 하는 장면을 카메라로 찍기 위해 장면상의 객체를 좌, 우(이동), 앞, 뒤(크기 변환)로 이동하고 회전하는 것

- 그래픽 소프트웨어에서의 기하학적 변환
  > 변환 행렬(4x4)에 의해서 구현

- 기하학적 변환은 장면에서의 한 개의 물체나 전체에 적용
  - 광역 변환: 전체의 축과 원점을 사용하여 특정 물체에 적용하는 변환
  - 지역 변환: 물체 자체의 축과 원점을 사용할 경우



3차원 변환

동차 좌표(homogeneous coordinate)

- 모조 좌표(dummy coordinate)를 추가하여 nxn 행렬로 연산

  4×4 행렬로 통일하기 위해 동차 좌표를 이용

모조 좌표 w를 추가하여 점을 p(x,y,z,w)로 표시

$$p = \begin{bmatrix} x \\ y \\ z \\ w \end{bmatrix} \Rightarrow p' = \begin{bmatrix} x' \\ y' \\ z' \\ w' \end{bmatrix}$$

초기에는 *w*=1로 설정

- 장점
  > 모든 변환을 동차 좌표의 행렬 곱으로 표현 가능
  > 변환 합성이 용이
  > 수치 계산의 감소
  > 고속 계산을 위한 병렬 처리 가능

비틀기

$$H_x(\theta,\varphi) = \begin{bmatrix} 1 & \cot\theta & \cot\varphi & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

평행 이동

$$T(\Delta x,\Delta y,\Delta z) = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

크기 변형

$$S(\alpha,\beta,\gamma) = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

회전

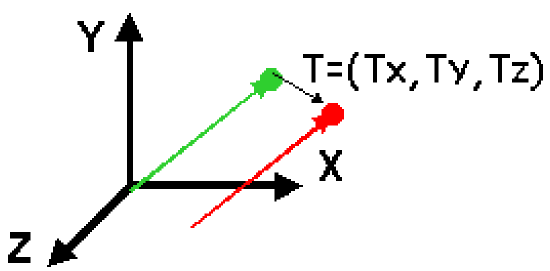
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

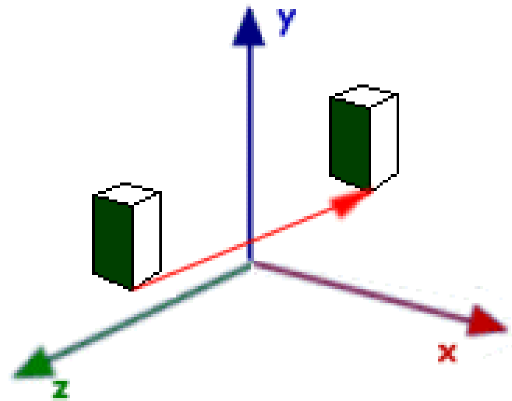
3차원 변환

이동 변환 (Translation)

- 3차원에서의 동질좌표 표현에서, 하나의 점이 (x, y, z) 지점에서 (x', y', z') 지점으로 위치 이동하는 것은 다음 행렬연산에 의하여 수행됨

$$[x' \ y' \ z' \ 1] = [x \ y \ z \ 1]\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ Tx & Ty & Tz & 1 \end{bmatrix} \qquad x' = x + Tx,\ y' = y + Ty,\ z' = z + Tz$$





- 3차원 물체의 이동변환:
  > 물체를 정의하는 모든 점에 대하여 이동변환 작업 수행
- 다각형에 대한 이동변환:
  > 각 면의 꼭지점에 대하여 이동변환 작업 수행

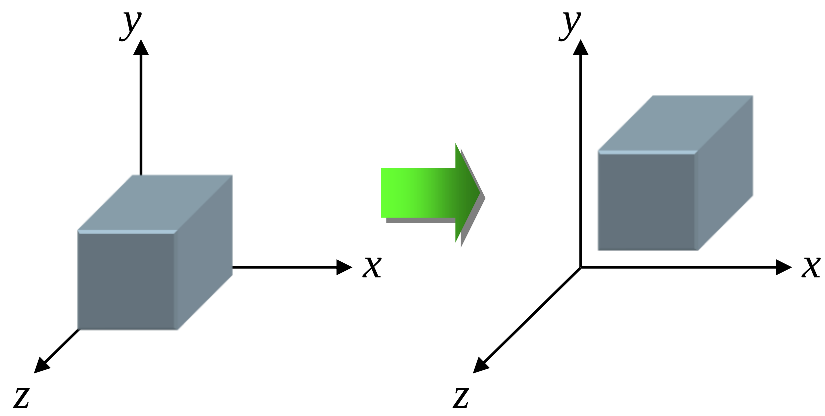
3차원 변환

평행 이동(Translation)

$$\begin{aligned} x' &= x + \Delta x \\ y' &= y + \Delta y \\ z' &= z + \Delta z \\ w' &= w \end{aligned}$$

$$T(\Delta x, \Delta y, \Delta z) = \begin{bmatrix} 1 & 0 & 0 & \Delta x \\ 0 & 1 & 0 & \Delta y \\ 0 & 0 & 1 & \Delta z \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

물체를 정의하는 모든 점에 대해
3개의 변수들을 이동



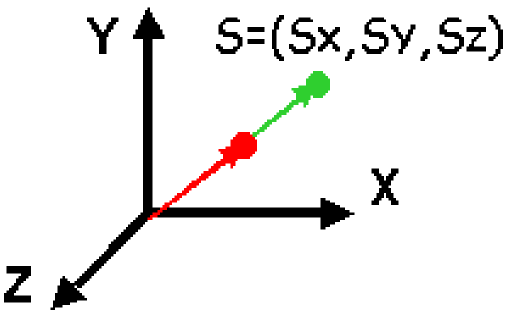
3차원 변환

크기 변환(Scaling)

- 물체의 크기 변경 방법
  > 현재의 비례를 그대로 유지하면서 크기 변환
  > 현재 물체의 비례 율과 관계 없이 크기 변경

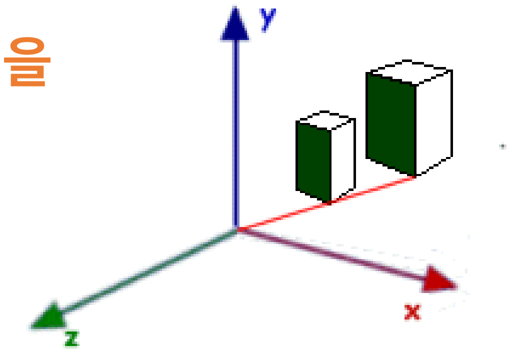
$$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1] \begin{bmatrix} S_x & 0 & 0 & 0 \\ 0 & S_y & 0 & 0 \\ 0 & 0 & S_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

$$x' = x \cdot S_x, \quad y' = y \cdot S_y, \quad z' = z \cdot S_z$$



변환행렬을 3차원 물체를 나타내는 각 점에 사용하면, 대상 물체는 좌표상의 원점을 기준으로 확대 / 축소됨

이 때 비율 값이 모두 같아야만 본래의 모습을 그대로 갖는다

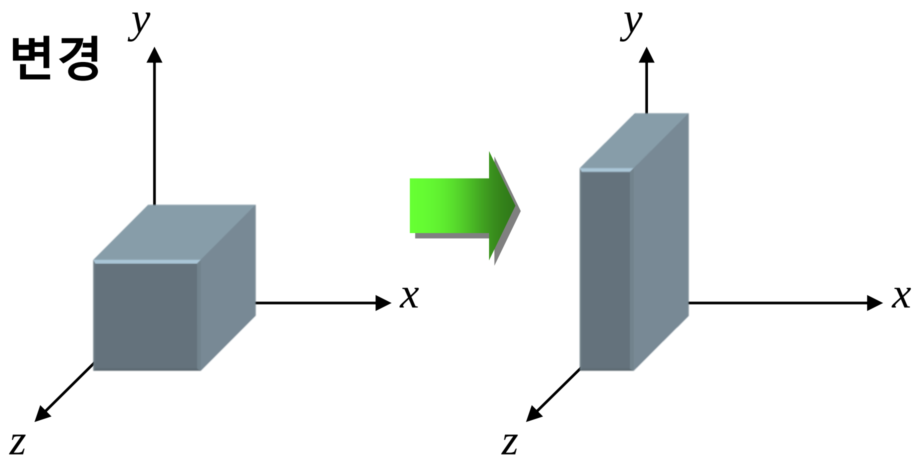


3차원 변환

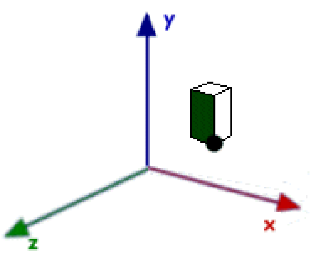
크기 변환(Scaling)

- 모든 3D에서 독립적인 확대 및 축소 변경

$$S(\alpha,\beta,\gamma) = \begin{bmatrix} \alpha & 0 & 0 & 0 \\ 0 & \beta & 0 & 0 \\ 0 & 0 & \gamma & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$



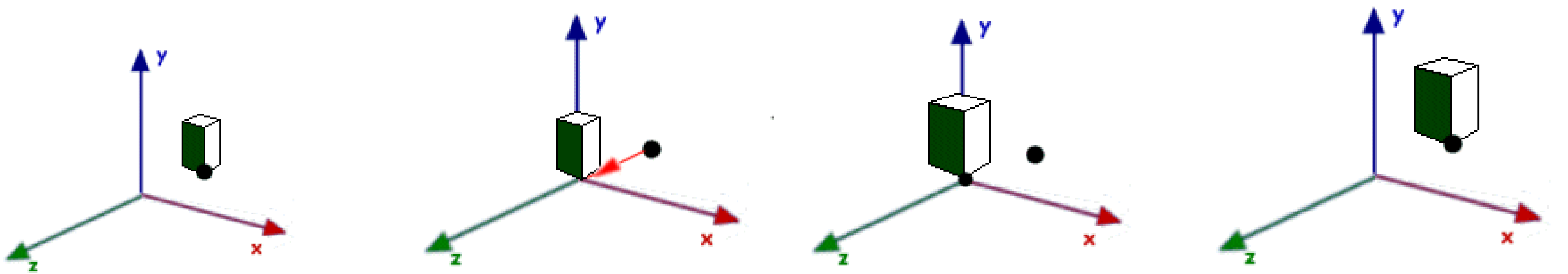
- 고정점($X_F Y_F, Z_F$)을 기준으로 확대 / 축소 하는 경우
  1. 고정 점을 원점 자리에 옮김
  2. 변환 행렬에 의하여 확대/축소 작업 수행
  3. 고정 점을 본래 위치로 환원



3차원 변환

크기 변환(Scaling)

- 고정점($X_F Y_F, Z_F$)을 기준으로 확대 / 축소 하는 경우
  1. 고정 점을 원점 자리에 옮김
  2. 변환 행렬에 의하여 확대/축소 작업 수행
  3. 고정 점을 본래 위치로 환원



- 1, 2, 3 단계를 나타내는 행렬의 곱

$$T(-x_F, -y_F, -z_F) \cdot S(Sx, Sy, Sz) \cdot T(x_F, y_F, z_F)$$

$$= \begin{bmatrix} Sx & 0 & 0 & 0 \\ 0 & Sy & 0 & 0 \\ 0 & 0 & Sz & 0 \\ (1-Sx)\, x_F & (1-Sy)\, y_F & (1-Sz)\, z_F & 1 \end{bmatrix}$$

3차원 변환

회전(Rotation)

- 현재 보이지 않는 물체의 다른 변을 보이게 하는데 사용

- 한 물체를 회전시킬 때, 반드시 물체가 어떤 축을 중심으로 회전하므로, 중심이 되는 회전축과 회전각도를 명시

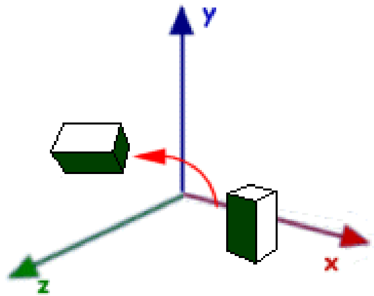
- 3D 회전의 경우, 회전축은 3차원 공간에서 임의의 방향이 될 수 있음
  > 회전축이 좌표축과 항상 평행
  > 3개의 좌표축을 적당히 회전시켜, 임의의 회전축에 따라 회전하는 결과

3차원 변환

z축, x축, y축이 회전축인 경우

z축을 회전축으로 회전

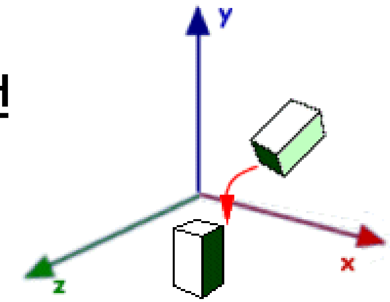
$$x' = x\cos\theta - y\sin\theta$$
$$y' = x\sin\theta + y\cos\theta$$
$$z' = z$$



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

x축을 회전축으로 회전

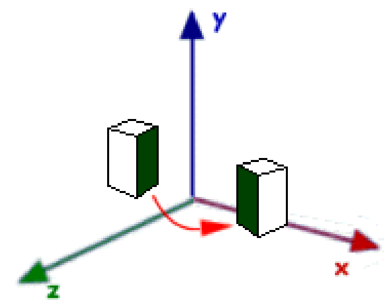
$$y' = y\cos\theta - z\sin\theta$$
$$z' = y\sin\theta + z\cos\theta$$
$$x' = x$$



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

y축을 회전축으로 회전

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

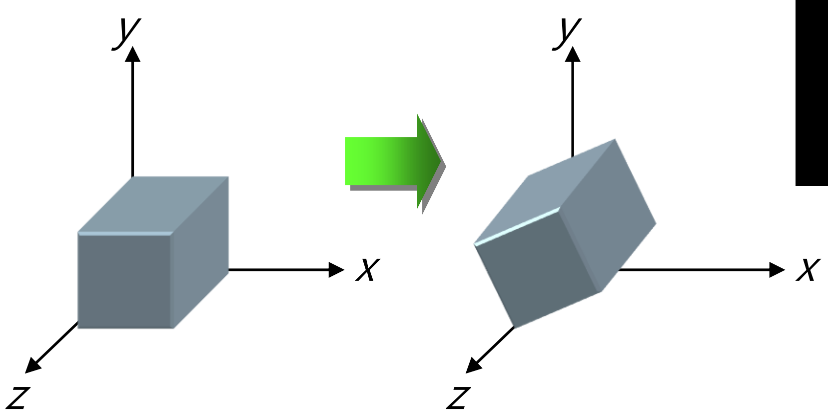


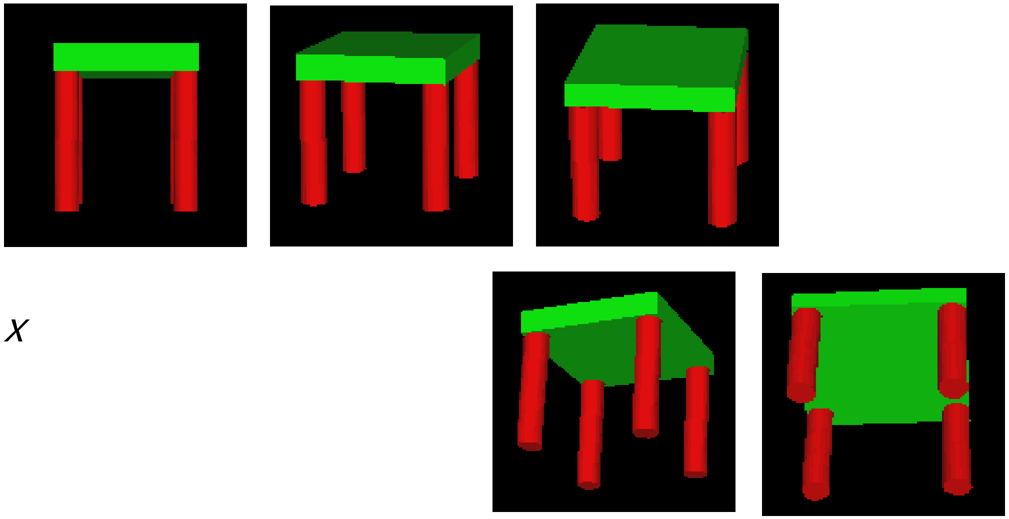
$$[x' \; y' \; z' \; 1] = [x \; y \; z \; 1]\begin{bmatrix} \cos\theta & 0 & -\sin\theta & 0 \\ 0 & 1 & 0 & 0 \\ \sin\theta & 0 & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3차원 변환

회전 변환 (Rotation)

- 3차원 회전은 각 축에 대해 독립적으로 이루어진다.





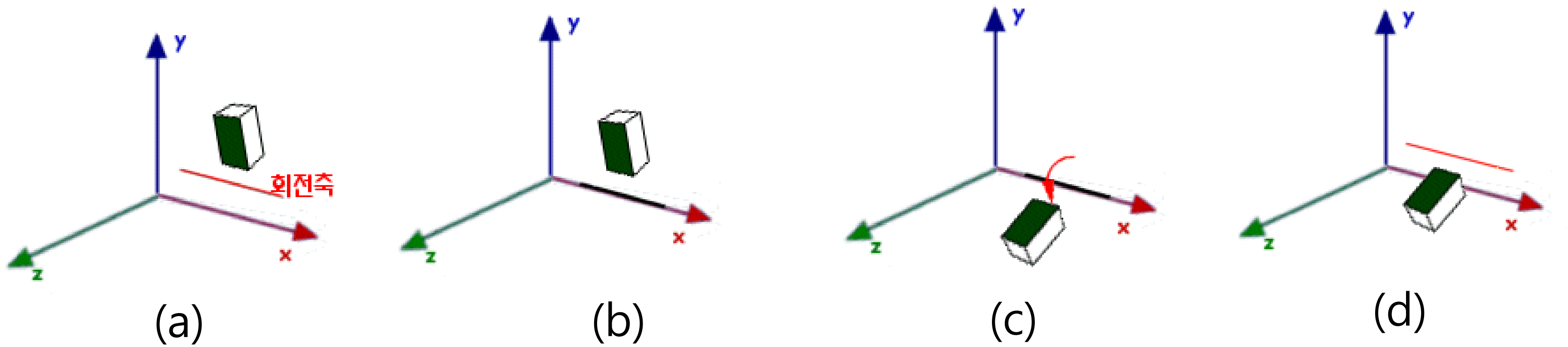
$$R_x(\theta) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

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

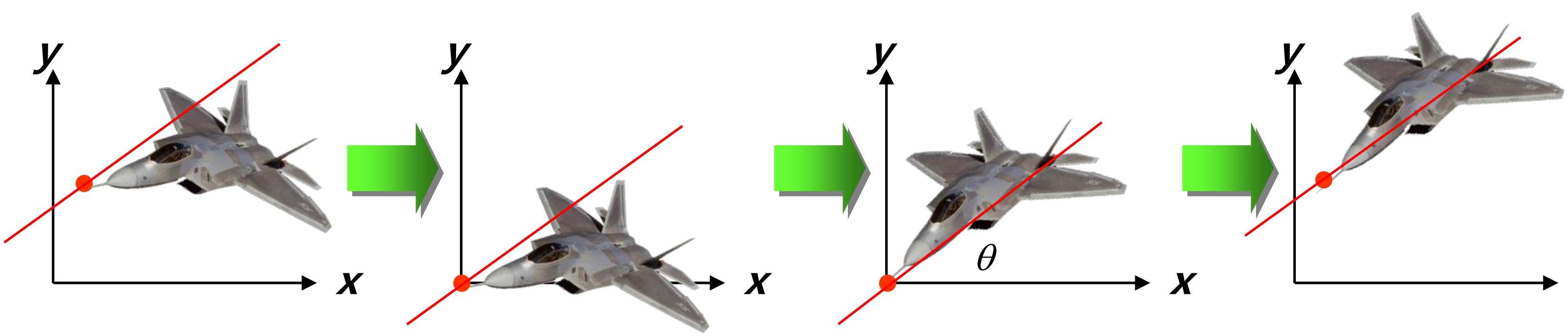
$$R_z(\theta) = \begin{bmatrix} \cos\theta & -\sin\theta & 0 & 0 \\ \sin\theta & \cos\theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3차원 변환

x 축과 평행인 임의 축을 기준으로 한 3D 회전



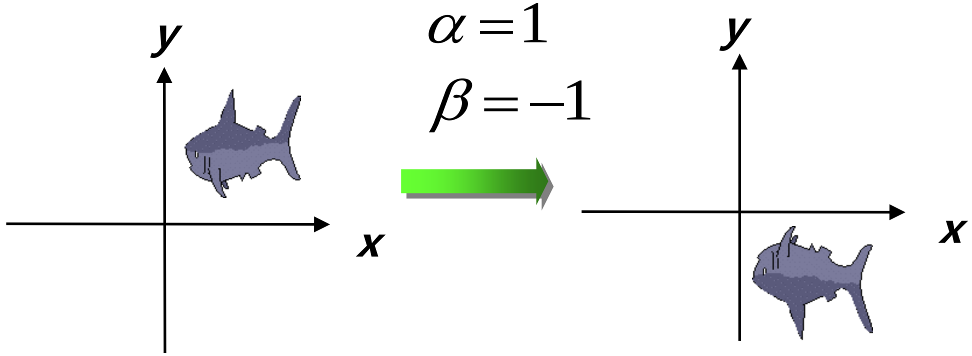
- 3차원 회전축이 xyz 축의 하나와 평행하는 경우



3차원 변환

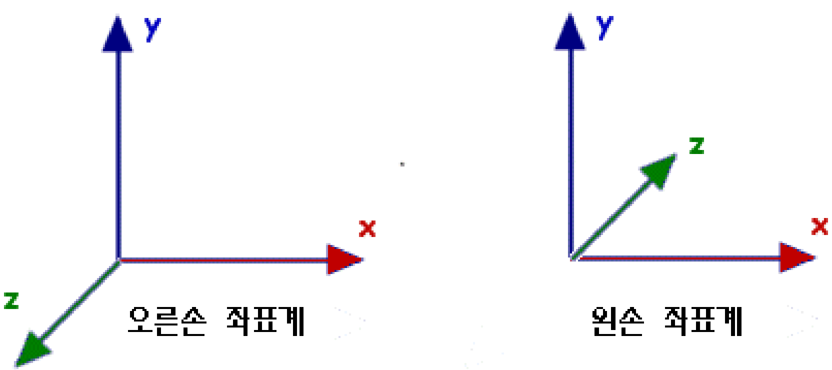
반사 변환 (Reflection Transformation)

- 3D에서는 반사는 특정 평면에 의하여 반사



$$\alpha = 1 \qquad \beta = -1$$

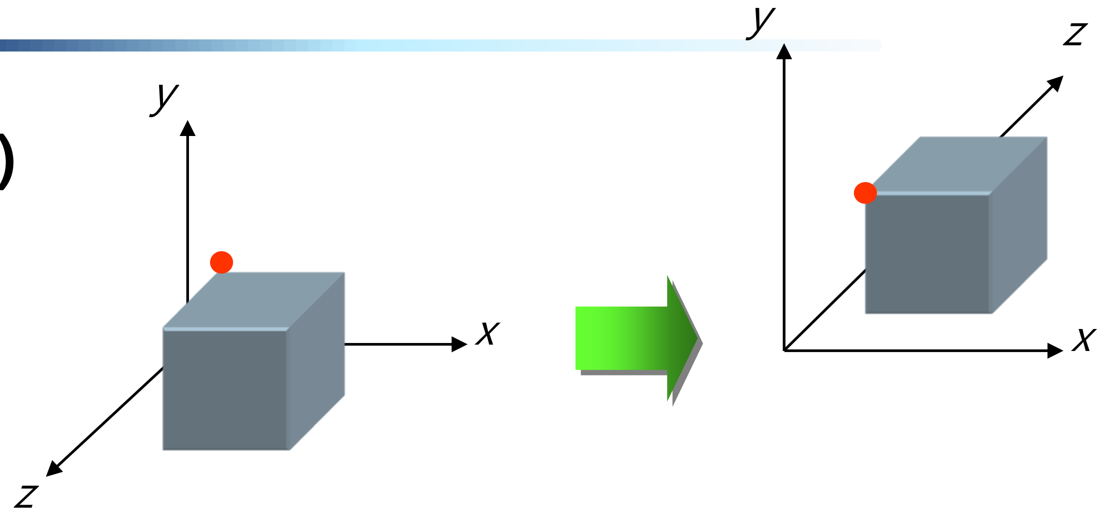
- 좌표의 값을 오른손 좌표체계에서 왼손 좌표체계(또는 그 반대 경우)로 바꾸는 경우



$$RFz = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

3차원 변환

반사(Reflection)



xy 평면에 대한 반사

$$\mathrm{Re}_{z=0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

xz 평면에 대한 반사

$$\mathrm{Re}_{y=0} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

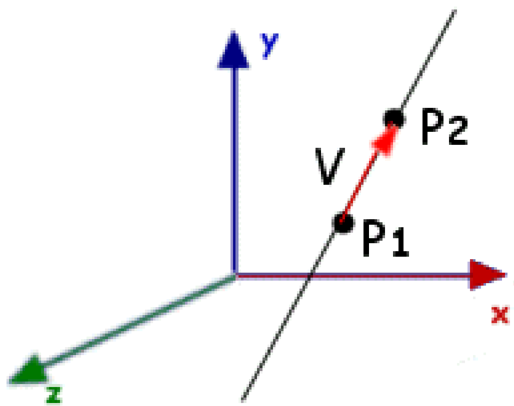
yz 평면에 대한 반사

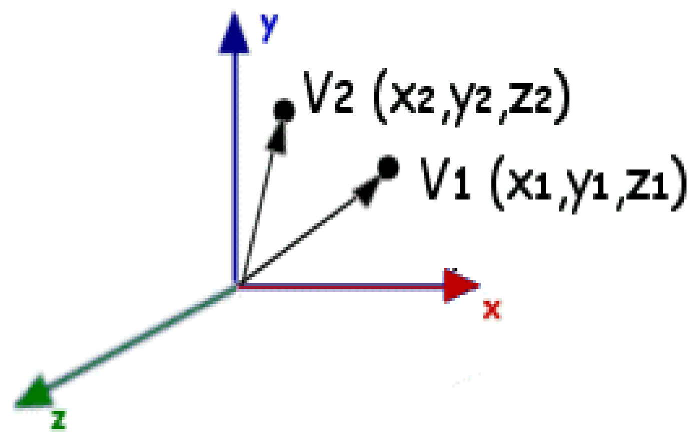
$$\mathrm{Re}_{x=0} = \begin{bmatrix} -1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$

벡터

벡터(Vector) 연산

- 벡터: 두 개의 점을 잇는 방향을 갖는 선분
- 하나의 선은 두 개의 점에 의해서 정해지므로, 회전축은 회전축에 따르는 벡터를 표시하여 나타냄





임의의 벡터를 출발점이 좌표축의 원점이 되도록 이동하게 되면 벡터를 좌표 상의 한 점으로 표현 가능

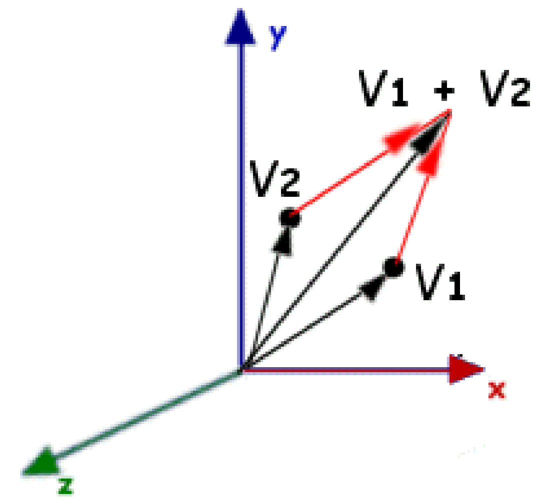
$$\overrightarrow{OV} = \vec{V} = (x_1, y_1, z_1)$$

벡터

벡터 덧셈과 곱셈

$$V_1 + V_2 = (x_1 + x_2,\ y_1 + y_2,\ z_1 + z_2)$$

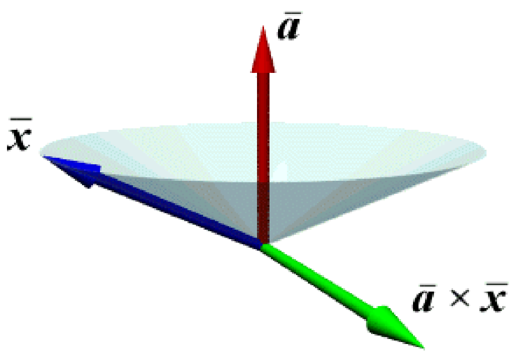
- 벡터 덧셈
  > 벡터로만 서로 더할 수 있음



- 벡터 곱셈
  > 벡터 내적: 계산 결과가 수치 값
  스칼라 곱, Scala Product, Inner Product

  > 벡터 외적: 계산 결과가 새로운 벡터
  벡터 곱, Vector Product, Outer Product

$$V_1 \cdot V_2 = x_1 x_2 + y_1 y_2 + z_1 z_2$$



$$V_1 \cdot V_2 = |V_1||V_2| \cos\theta$$

$$|V| = \sqrt{V \cdot V} = \sqrt{x^2 + y^2 + z^2}$$

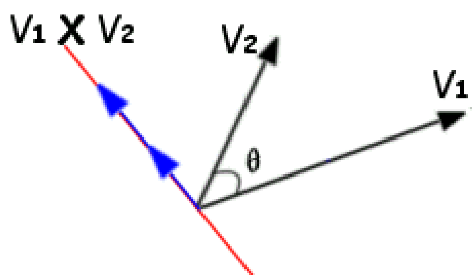
벡터

벡터 곱(외적)

$$\begin{vmatrix} a & b & c \\ d & e & f \\ g & h & i \end{vmatrix} = +a(ei - fh) - b(di - fg) + c(dh - eg)$$

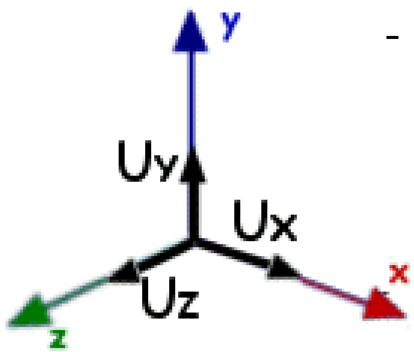
- 벡터 곱 V1 X V2 계산 결과 두 개의 벡터 모두에 수직인 또 하나의 다른 벡터 생성

- 벡터 곱에 의해 생기는 벡터의 방향은 단위벡터 u 에 의해서 표시



단위 벡터 u 의 방향은 벡터 V_1에서 벡터 V_2의 방향으로 오른손을 사용하여 수직인 가상의 선에서 엄지손가락이 가르키는 방향이 단위벡터 u의 방향이 된다.

$$V_1 \times V_2 = u|V_1||V_2|\sin\theta$$



- 각 좌표축에 평행인 단위 벡터는 행렬식의 결과와 일치

$$(y_1z_2 - z_1y_2,\ \ z_1x_2 - x_1z_2,\ \ x_1y_2 - y_1x_2)$$

$$V_1 \times V_2 = \begin{vmatrix} u_x & u_y & u_z \\ x_1 & y_1 & z_1 \\ x_2 & y_2 & z_2 \end{vmatrix}$$