OpenGL 4

가상현실론

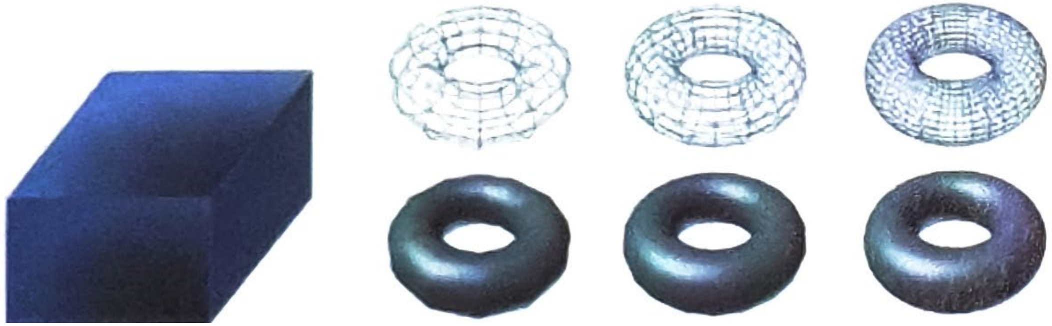
2021/03/10

목차

- 모델변환과 시점변환
  - 좌표계
  - 기하변환
  - GL의 모델 변환

3차원 물체 표현

- 경계면 표현(boundary surface representaiton): 그래픽에서 물체 표면만 표현하며 물체면을 평면 다각형 (planar polygon)의 집합으로 나타냄. 곡면(curved surfaces) 역시 최종적으로는 작은 다각형의 집합으로 표현.

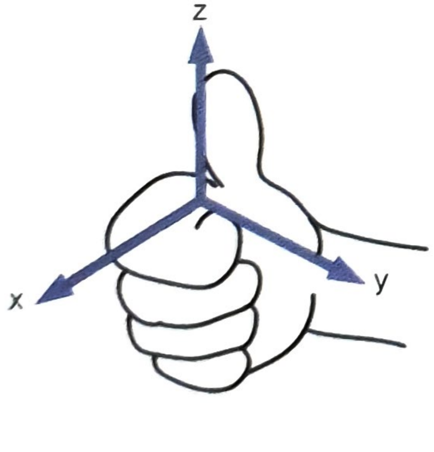


- 육면체는 6개의 평면 다각형만으로 표현할 수 있으나 오른쪽 그림과 같은 torus 곡면은 훨씬 많은 다각형으로 표현해야 함. 더 많은 다각형은 더 부드러운 곡면을 표현할 수 있으나 연산량 또한 증가

3차원 물체 표현

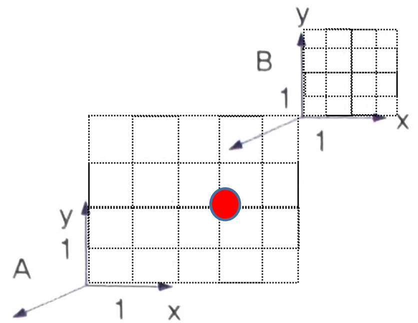
- 곡면을 표현하기 위해 필요한 다각형의 수는 곡면의 곡률(curvature)에 정비례. 곡면을 표현하는 평면 다각형 하나하나를 메쉬(mesh), 표면 메쉬(surface mesh), 다각형 메쉬(polygon mesh), 표면 다각형(surface polygon), 또는 단순히 다각형(polygon)이라고 부름.
- 일반적으로 표면 메쉬로 사용되는 다강형은 사각형이나 삼각형 사각형 메쉬(rectangular mesh)는 삼각형 메쉬(triangular mesh)에 비해 정밀도가 떨어짐.

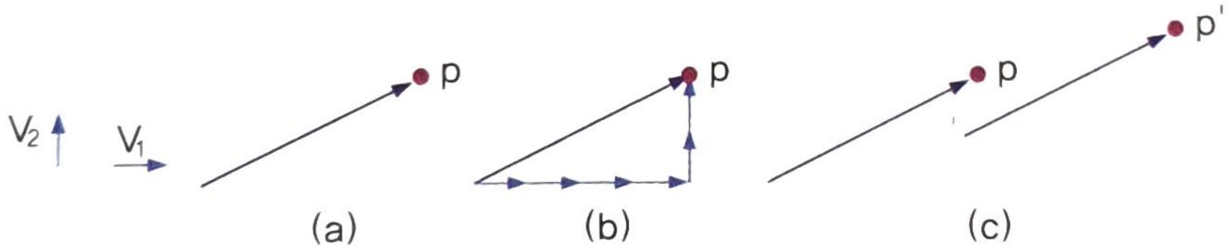
좌표축과 좌표계



- 3차원 공간에서 물체의 위치는 주어진 좌표계(coordinate system)을 기준으로 표시. 가장 많이 사용되는 좌표계는 직교 좌표계 (Cartesian coordinate system).
- 3차원 직교좌표계는 그림과 같이 원점에서 서로 직각으로 교차하는 3개의 좌표축 벡터(coordinate axis vector)로 이루어짐. x, y, z축의 방향은 일반적으로 오른손 법칙을 따름. 컴퓨터 그래픽에서는 일반적으로 오른속 법칙을 따름.

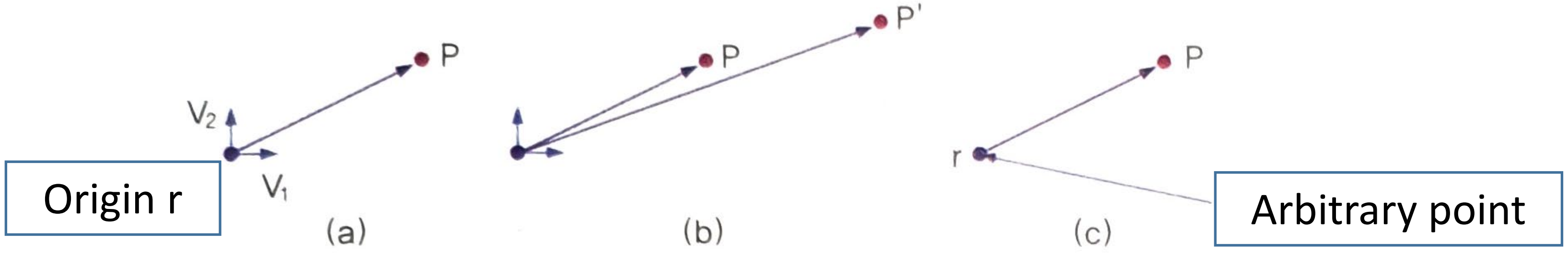
직교좌표계, basis vector





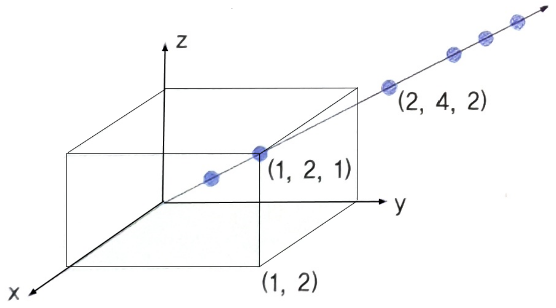
- 직교좌표계는 원점 위치, 축 방향, 축 눈금의 길이 등에 의해 정의. 또 점의 좌표는 기준이 되는 좌표계에 따라서 달라짐
- 그림의 붉은 점은 A 좌표계를 기준으로 하면 (3, 2, 0)이지만, B 좌표계를 기준으로 하면 (-4, -4, 0)임.
- V_1과 V_2처럼 자신의 합성에 의해 따른 모든 벡터를 표현할 수 있는 벡터를 기반 벡터(basis vector)라고 함. 기반 벡터의 특징은 그들끼리 linear independent하다는 점. 시각적으로 공간상에서 서로 직교하는 벡터는 linearly indendent.

차원 (dimension)



- 차원(dimension): 점의 위치를 표현하기 위한 기반 벡터의 수
- 어파인 공간(affine space): 벡터+점
- 어파인 공간에서는 기반 벡터를 한점(원점, origin)에 고정함.
- 3차원 정점 좌표를 나타내기 위해서는 원점과 3개의 basis vector가 필요함. 예를 들어 3차원 좌표계는 (r, V_1, V_2, V_3)로 표현할 수 있음.
- 예) $v = 4V_1 + 2V_2 + V_3$
- $p = r + 4V_1 + 2V_2 + V_3$

동차좌표(homogeneous coordinate)



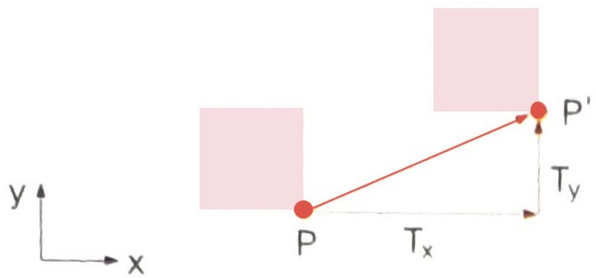
- 동차좌표는 3차원 요소(점, 벡터)를 4차원으로 표현

$$v = 4V_1 + 2V_2 + V_3 + 0 \cdot r$$
$$P = 4V_1 + 2V_2 + V_3 + 1 \cdot r$$

- 마지막 요소가 0이면 벡터를 1이면 점으로 표현
- 3차원 동차좌표를 4차원 (x, y, z, w)로 표현하면 3차원 실제 좌표는 (x/w, y/w, z/w)
- 컴퓨터 그래픽에서는 모든 좌표를 동차 좌표로 표시
- 하드웨어 또한 동차 좌표 처리를 위해 한 번에 4개의 요소를 입력하고 처리할 수 있도록 설계

기하변환 (Geometric Transformation)

- 기하 변환 (geometric transformaiton or transformation: 물체의 위치를 옮기거나 모습을 변형 (이동, 회전, 크기조절)



- 이동 변환 (translational transformation): 물체를 이동하는 변환

$$P' = T + P$$

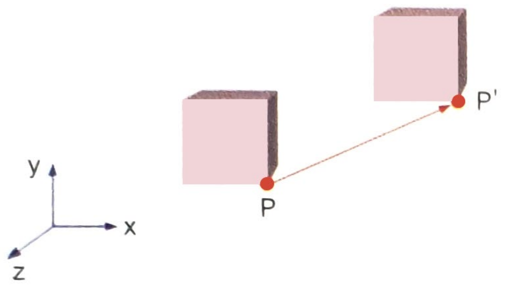
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix}$$

- T_x와 T_y는 각각 x축과 y축으로 이동한 양.
- 하지만 그래픽에서는 덧셈 표현이 사용되지 않음.

이동변환 (Translation Tranformation)

- 문제의 답은 동차 좌표 (homogeneous coordinate system)

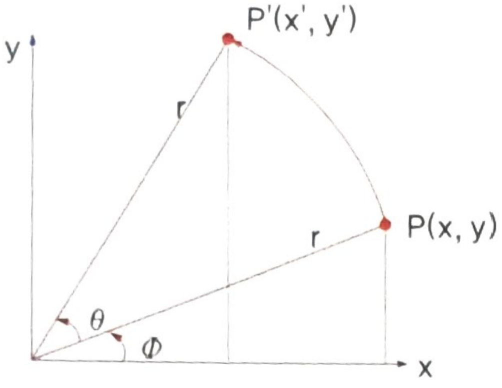
$$\begin{bmatrix} x' \\ y' \end{bmatrix} = \begin{bmatrix} T_x \\ T_y \end{bmatrix} + \begin{bmatrix} x \\ y \end{bmatrix} \quad \text{(X)} \qquad \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & T_x \\ 0 & 0 & T_y \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \text{(O)}$$



$$P' = TP$$

$$\begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 0 \end{bmatrix}$$

회전 (Rotation Transformation)



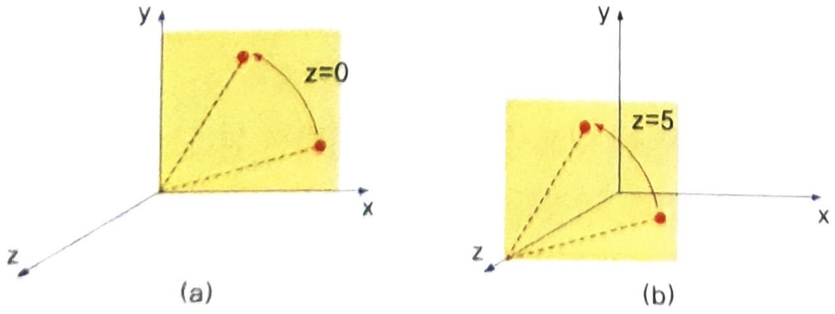
2차원 회전

x’ = r cos(ϕ+θ)=r cos ϕ cos θ- r sin ϕ sin θ=x cos θ-y sin θ

Y’ = r sin(ϕ+θ)=r cos ϕ sin θ+ r sin ϕ cos θ=x sin θ+y cos θ

P’=R·P

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

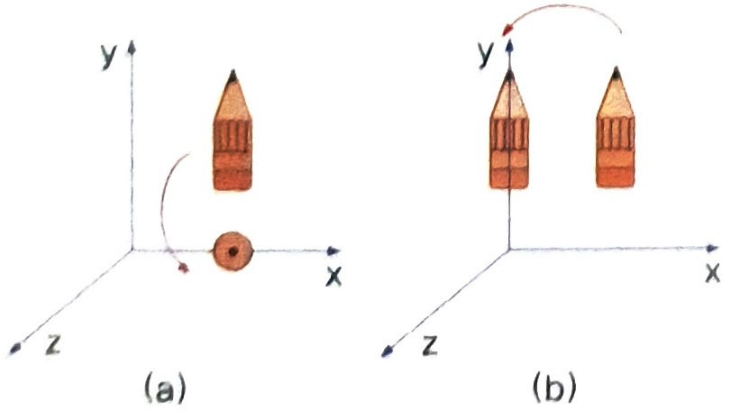


3차원 회전

P’=$\mathrm{R_z}(\theta)$·F

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

회전 (Rotation Transformation)



x-축 회전 y-축 회전

- x-축 중심 회전

P'=$R_x(\theta)$·F

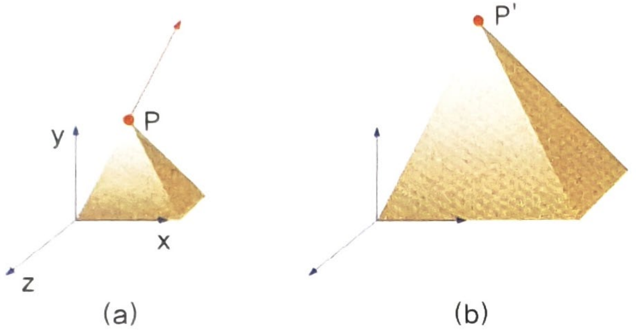
$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos\theta & -\sin\theta & 0 \\ 0 & \sin\theta & \cos\theta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

- y-축 중심 회전

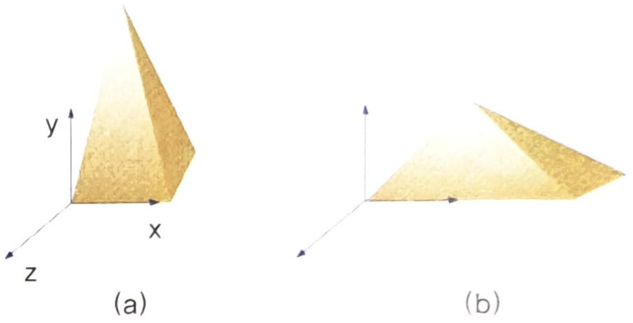
P'=$R_y(\theta)$·F

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

크기 조절 (Scaling Transformation)



균등 크기 조절



차등 크기 조절

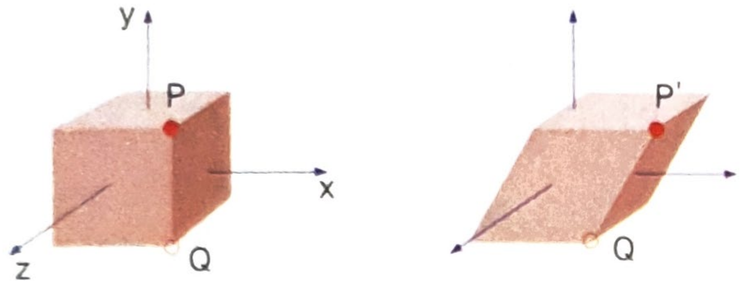
- 좌표계 원점을 기준으로 물체의 크기 조절

P’=S·P

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

- S_x, S_y, S_z는 각각 x,y,z축 방향의 배율.
- 배율이 1보다 크면 확대, 작으면 축소.
- 모든 배율이 같으면 균등 크기 조절(Uniform scaling)
- 하나라도 배율이 다르면 차등 크기 조절(differential scaling)

전단 (Shear Transformation)



(a) (b)

- 선난(Shearing) : 불제들 한쏙 방향으로 밀어내서 변형시킨 것
- 그림 (a), (b)의 예의 경우 P는 P’으로 옮겨갔지만 Q는 원래 위치에 존재.

$$x' = x + Sh_y \cdot y$$
$$y' = y + Sh_x \cdot x$$

- Sh_x, Sh_y는 각각 x, y 방향의 전단 인수(shearing factor)
- x-y 평면을 따라 가해진 전단 변환 행렬식은 아래와 같음

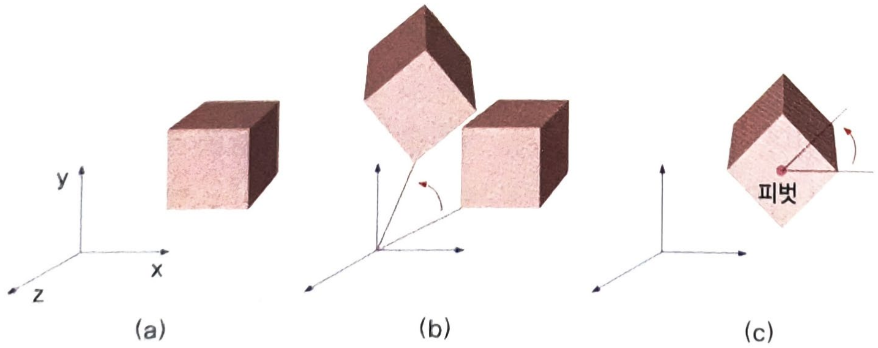
$$P' = Sh \cdot P$$

$$\begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & Sh_y & 0 & 0 \\ Sh_x & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}$$

복합변환 (Composite Transformation)

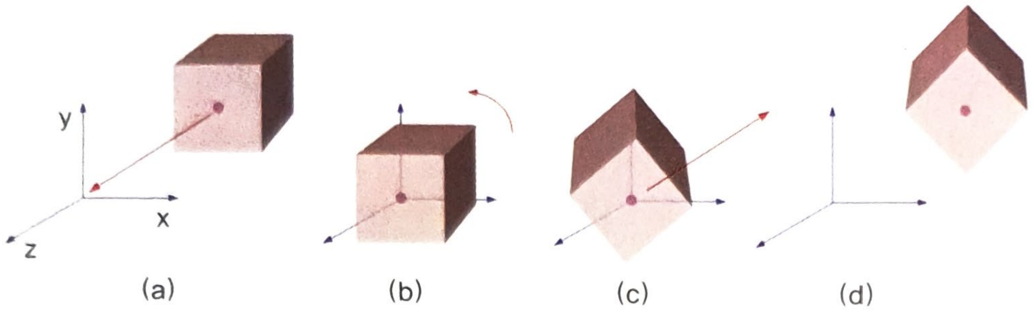
- 복합행렬 (Composite Matrix): C와 같이 연속된 변환 행렬을 모두 곱하여 하나의 행렬로 표시한 것
- 복합변환 (Composite Transformation): 복합행렬로 표현되는 일련의 연속된 변환
- 예)물체에 대해서 크기 조절 변환(S1)을 가한 후 회전(R1)한 후 다시 크기 조절(S2). 새로운 정점의 좌표는
- P ' =S2·R1·S1·P(<- 작업이 가해진 순서와 역순임에 주의)
- P’=C ·P (C= S2·R1·S1)
- 중심점 기준 회전 (Pivot Point Rotation): 원점이 아닌 물체의 중심점(pivot point)를 기준으로 회전

복합변환 (Composite Transformation)



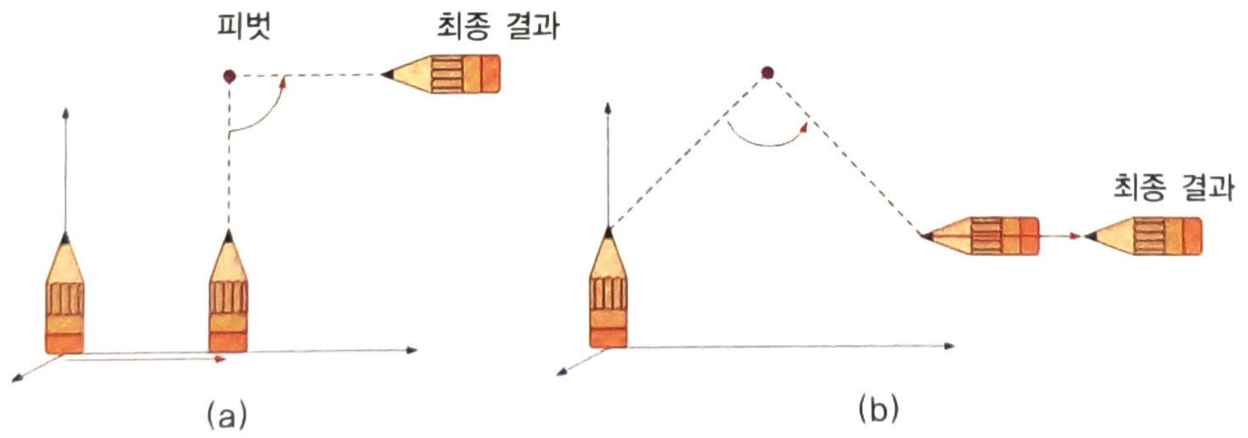
- z축 중심의 회전(b)와 피벗 중심의 회전(c)는 다른 결과를 가져옴
- 피벗 회전은 아래와 같은 3가지 변환이 복합되어 실행
  1. 피벗이 좌표계 원점에 일치하도록 물체를 이동
  2. 물체를 원점 중심으로 기준 축 주위로 회전
  3. 회전된 물체를 1에서 이동한 방향의 반대 방향으로 이동

복합변환-중심축 기준 회전



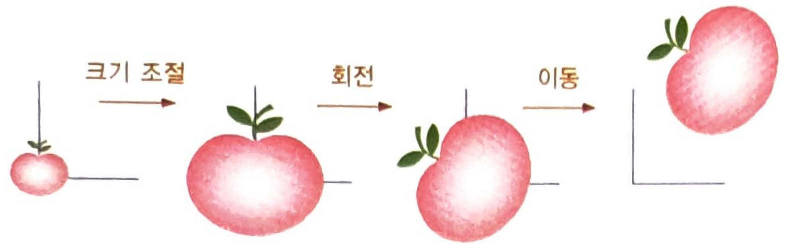
- (a)-(b): 물체의 중심을 원점에 일치시킴
- (b)-(c): z 축 중심 회전
- (c)-(d): 물체의 중심점으로 다시 이동
- $\mathrm{C} = \left(\mathrm{X_p}, \mathrm{Y_p}, \mathrm{Z_p}\right) \cdot \mathrm{R_z}(\theta) \cdot \left(\text{-}\mathrm{X_p}, \text{-}\mathrm{Y_p}, \text{-}\mathrm{Z_p}\right)$

복합변환



- 행렬의 곱셈에 교환법칙은 성립하지 않음에 유의 ($A \cdot B \neq B \cdot A$)
- (a)는 이동 후 회전을 가한 것이며 (b)는 회전 후 이동을 가한 것. (a)와 (b)의 경우 회전을 위한 피벗의 위치는 동일하나 결과는 다르다는 것에 유의

복합변환



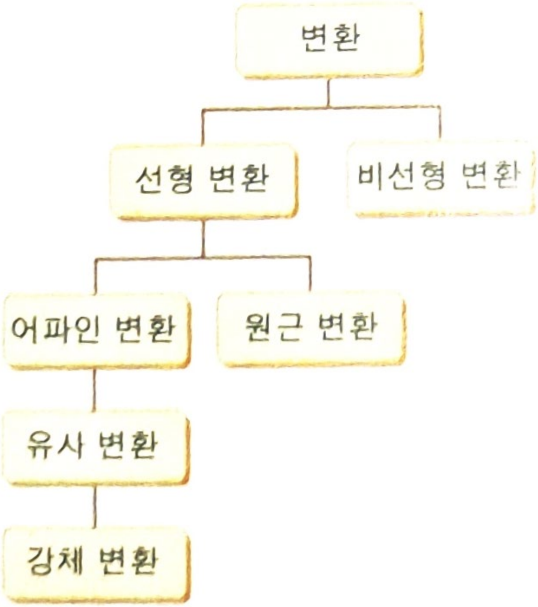
[그림 6-31] 크기 조절, 회전, 이동

- 변환된 물체 하나를 원래 물체의 인스턴스(instance)라고 함
- 물체 인스턴스는 크기 조절, 회전 이동 순으로 변환을 가함
- 크기 조절에이나 회전을 먼저 가하는 이유는 이 상태에서 물체 중심이 피벗과 원점이 일치한 상태이므로 이를 일치시키기 위한 변환이 불필요하기 때문.
- C=T·R·S

변환의 분류

- 강체 변환(Rigid Body Transformation)
  - 이동변환+ 회전변환
  - 변환 전후에 내부 정점 간의 거리가 그대로 유지됨
- 유사변환 (Similarity Transformation)
  - 강체변환+균등 크기 조절 변환+반사 변환
  - 변환 전후에 물체면 사이의 각이 유지되고, 물체 내부 정점 간의 거리도 일정한 비율로 유지됨 (크기 조절 변환 때문에 거리 자체가 유지되는 것은 아니라는 데 유의)
- 어파인 변환(Affine Transformation)
  - 유사변환+차등 크기 조절 변환+전단 변환
  - 공학이나 자연과학 문제 해결에 사용되는 거의 모든 변환
  - 물체의 타입이 유지 (직선-> 직선, 다각형-> 다각형, 곡면 -> 곡면)
  - 물체 내부의 평행한 선분이 변환 후에도 평행하게 유지
  - 행렬의 마지막 행이 항상 (0, 0, 0, 1)

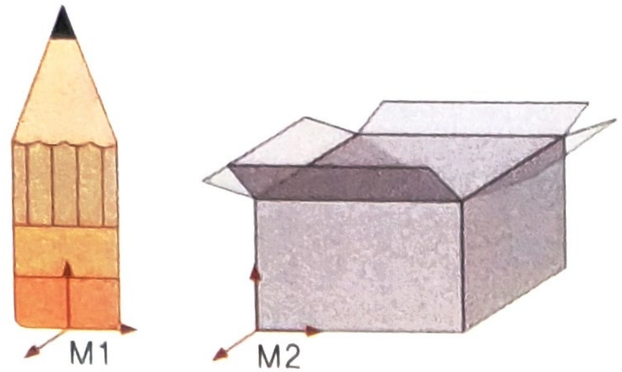
변환의 분류



- 원근 변환 (Perspective Transformation)
  - 직선이 직선으로 변환된다는 정도의 속성만 유지
  - 변환 행렬의 마지막 행이 (0, 0, 0, 1)이 아님
- 선형 변환(Linear Transformation)
  - 어파인 변환+원근 변환
  - 선형 조합(linear combination)으로 표시되는 변환

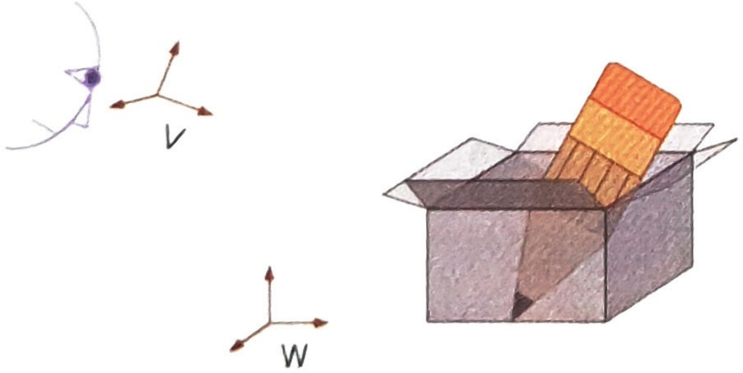
GL의 모델 변환

- 물체 정점은 물체 하나를 설계할 때의 좌표계, 한 장면에 여러 물체를 모아놓았을 때의 좌표계, 그 장면을 바라보는 시점에 따른 좌표계 등의 좌표계를 거치면서 새로운 좌표값으로 바뀌며 최종적으로 화면에 그려짐.
- 그래픽에서 모델링은 물체 정점을 결정하는 작업. 정점을 조합하여 다각형 메쉬를 만들고, 다각형 메쉬를 조합하여 물체를 만들고, 다시 물체를 조합하여 장면(Scene)을 구성할 수 있기 때문.



- 물체 좌표계(MCS: Modelling Coordinate System)/지역 좌표계(LCS: Local Coordinate System): 각 물체별로 설계상의 편의를 위주로 설정된 좌표계

GL의 모델 변환



- 연필과 박스가 각각 M1, M2의 지역좌표계를 가지고 있기 때문에 모든 물체를 한꺼번에 아우를 수 있는 기준 좌표계가 필요. 이를 전역 좌표계 (WCS: World Coordinate System)이라고 함.
- 그림에서는 W를 전역좌표계로 설정. 전역 좌표계가 설정되면 각 모델 좌표계를 기준으로 표시된 물체 좌표가 전역 좌표계를 기준으로 바뀌어 물체 간의 상대적인 위치가 명확해짐.
- 화면에 보이는 것은 시점(Viewpoint)가 좌우함. 사용자의 시점을 기준으로하는 시점 좌표계(VCS: View Coordinate System)으로 WCS가 다시 변환됨 (그림에서 V).