# 컴퓨터 그래픽스
## 제7장 3차원 그래픽스의 기하변환과 뷰잉

2019년 2학기

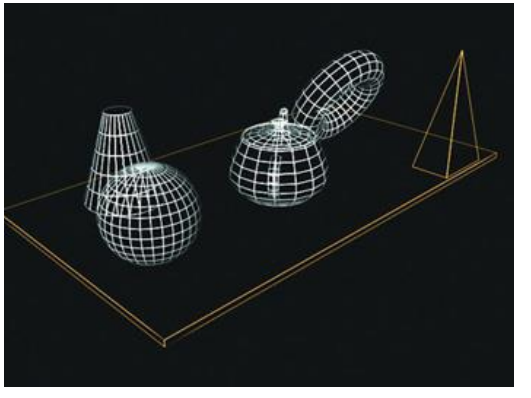
# 7장 학습 내용

- **3차원 그래픽스 기하 변환과 뷰잉**
  - 3차원 그래픽스의 처리 과정
  - 3차원 기하 변환
  - 투영
  - 좌표계 변환

# 3차원 그래픽스의 처리과정

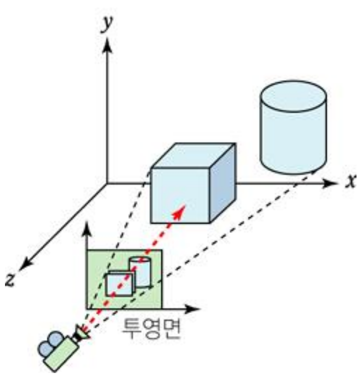
- **3차원 그래픽스**
  - 모델링 과정 (Modeling): 3차원 객체들을 형상화
  - 투영 과정 (Projection): 모델링한 객체들을 모니터와 같은 2차원 평면에 투영
  - 렌더링 과정 (Rendering): 객체에 현실감을 부여

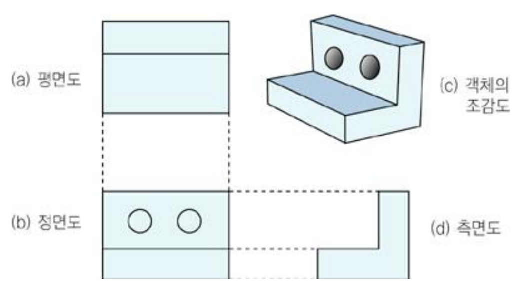
- **모델링 (Modeling, 3D Object Representation)**
  - 다각형 표면 모델링 (Polygon Surface Modeling)
  - 매개변수를 이용한 곡면 모델링 (Parametric Surface Modeling)
  - 와이어 프레임 (Wire-frame)
  - 솔리드 모델링 (Solid Modeling)
  - 스위핑 (Sweeping)
  - 프랙털 기하학 (Fractal Geometry)
  - 입자 시스템 (Particle System)

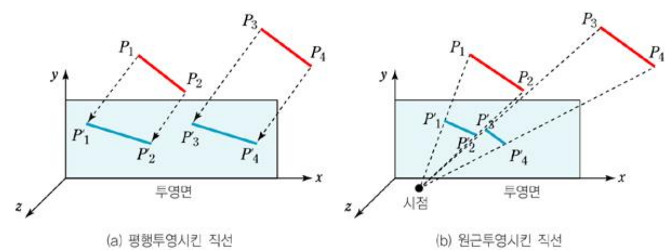


# 3차원 그래픽스의 처리과정

- Projection (투영)
  - 3차원 공간의 원뿔 -> 2차원의 삼각형 모양
  - Parallel Projection (평행투영)
    - 객체를 구성하는 각 요소들간의 상대적인 크기가 보존된다
    - 기계 설계, 건축 설계
  - Perspective Projection (원근투영)
    - 객체의 원근감이 잘 나타난다
    - 투영면에 보이는 2차원 객체의 크기는 3차원 객체와 투영면과의 거리에 반비례한다.
    - 건물의 조감도

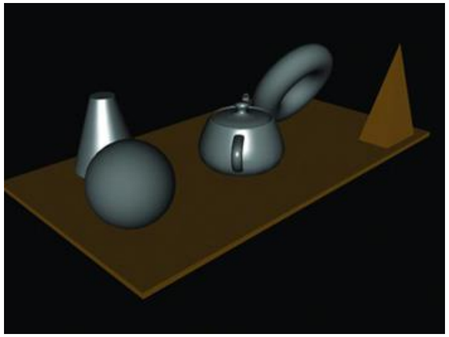
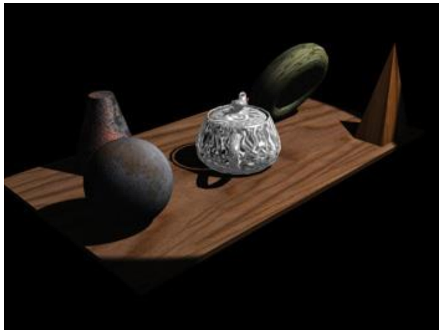


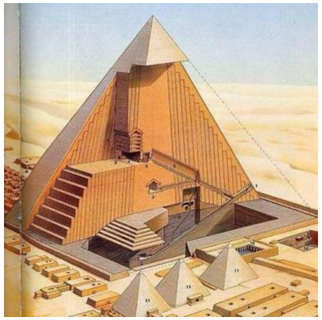



# 3차원 그래픽스의 처리과정

- Rendering (렌더링)
  - 투영된 그림을 렌더링하고 그림자나 색상의 변화를 표현하여 현실감 있는 그림을 만들어낸다.
    - Hidden Surface Removal (은면제거)
    - Shading (쉐이딩)
    - Texture Mapping (텍스쳐 매핑)
  - Depth Cueing  (깊이 표시법)
    - 와이어프레임 객체를 현실감 있게 표시한다.
    - 시점에서 객체까지의 거리에 따라 와이어의 밝기를 다르게 나타내어 현실감을 추구한다.
    - 가까운 부분은 밝게, 먼 부분은 어둡게 표현
  - Exploded View (Cutaway View, 분해도, 단면도)
    - 3차원 객체를 표현할 때 객체의 내부 또는 절단면을 보여주면 객체의 내부구조를 정확하게 파악할 수 있다

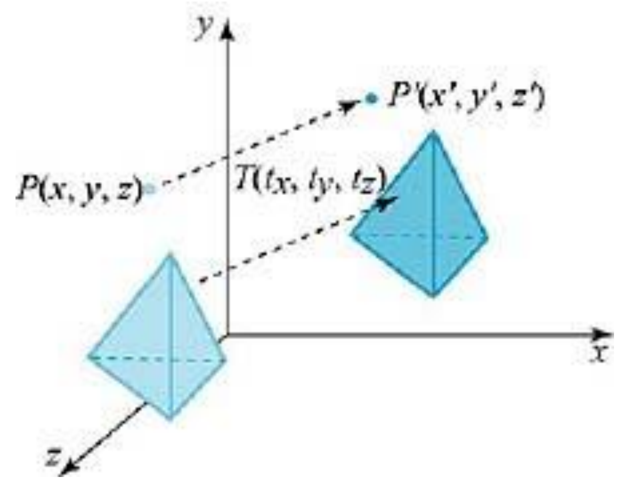






# 3차원 기하변환: 이동

- Translation (이동)
  - 공간상의 한 점에 이동거리를 더하여 이동한다.
  - $x' = x + t_x$ $y' = y + t_y$ $z' = z + t_z$
  - $P' = T(t_x, t_y, t_z) \bullet P$ $(t_x, t_y, t_z)$: 이동 벡터

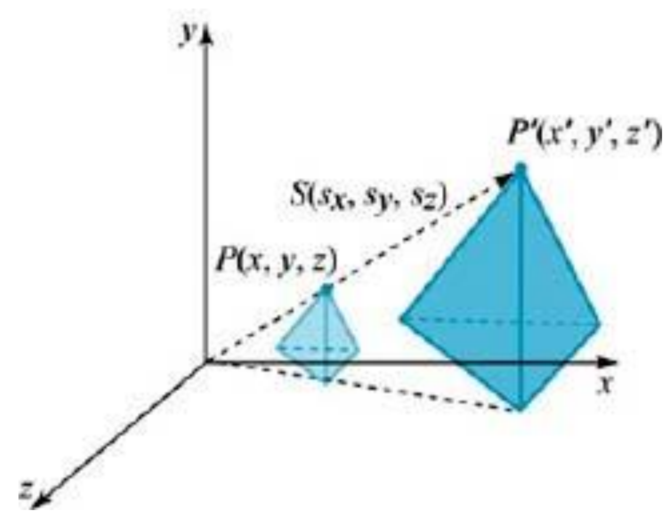
$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & t_x \\ 0 & 1 & 0 & t_y \\ 0 & 0 & 1 & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = T(t_x, t_y, t_z) \cdot P$$



# 3차원 기하변환: 신축

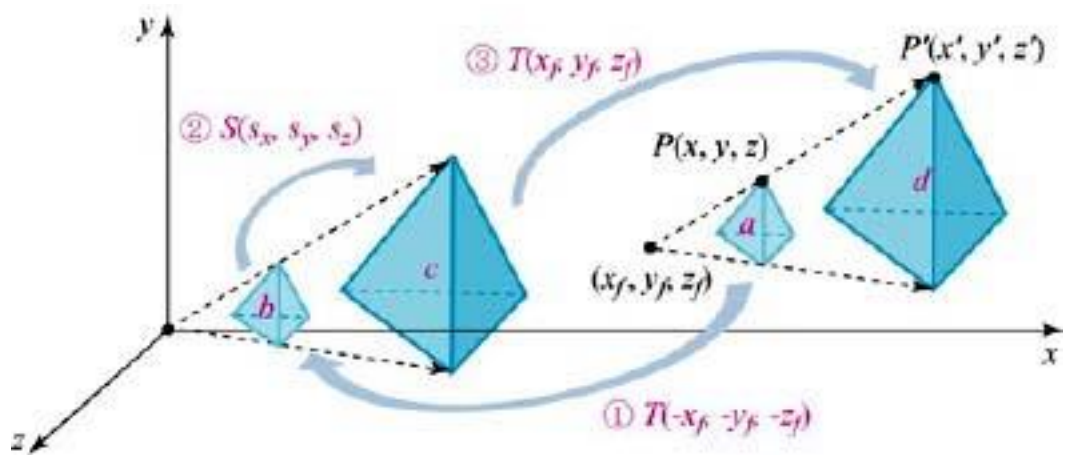
- Scaling (신축)
  - 객체의 크기를 확대 또는 축소, 비율 변화
  - $x' = s_x \bullet x$ $y' = s_y \bullet y$ $z' = s_z \bullet z$
  - $P' = S(s_x, s_y, s_z) \bullet P$ $(s_x, s_y, s_z)$: 신축률

$$P' = \begin{bmatrix} x' \\ y' \\ z' \\ 1 \end{bmatrix} = \begin{bmatrix} s_x & 0 & 0 & 0 \\ 0 & s_y & 0 & 0 \\ 0 & 0 & s_z & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} = S(s_x, s_y, s_z) \cdot P$$



# 3차원 기하변환: 신축

– 임의의 점에 대한 신축
  - Translation: $P' = (x_f, y_f, z_f) \bullet P$
  - Scaling: $P'' = (s_x, s_y, s_z) \bullet P'$
  - Translation: $P''' = (-x_f, -y_f, -z_f) \bullet P''$



# 3차원 기하변환: 회전

- Rotation (회전)
  - 객체를 기준 축 주위로 회전시킨다.
  - z-축 회전: $P' = R_z(\theta) \bullet P$

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

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

    $z' = z$
  - x-축 회전 : $P' = R_x(\theta) \bullet P$

    $x' = x$

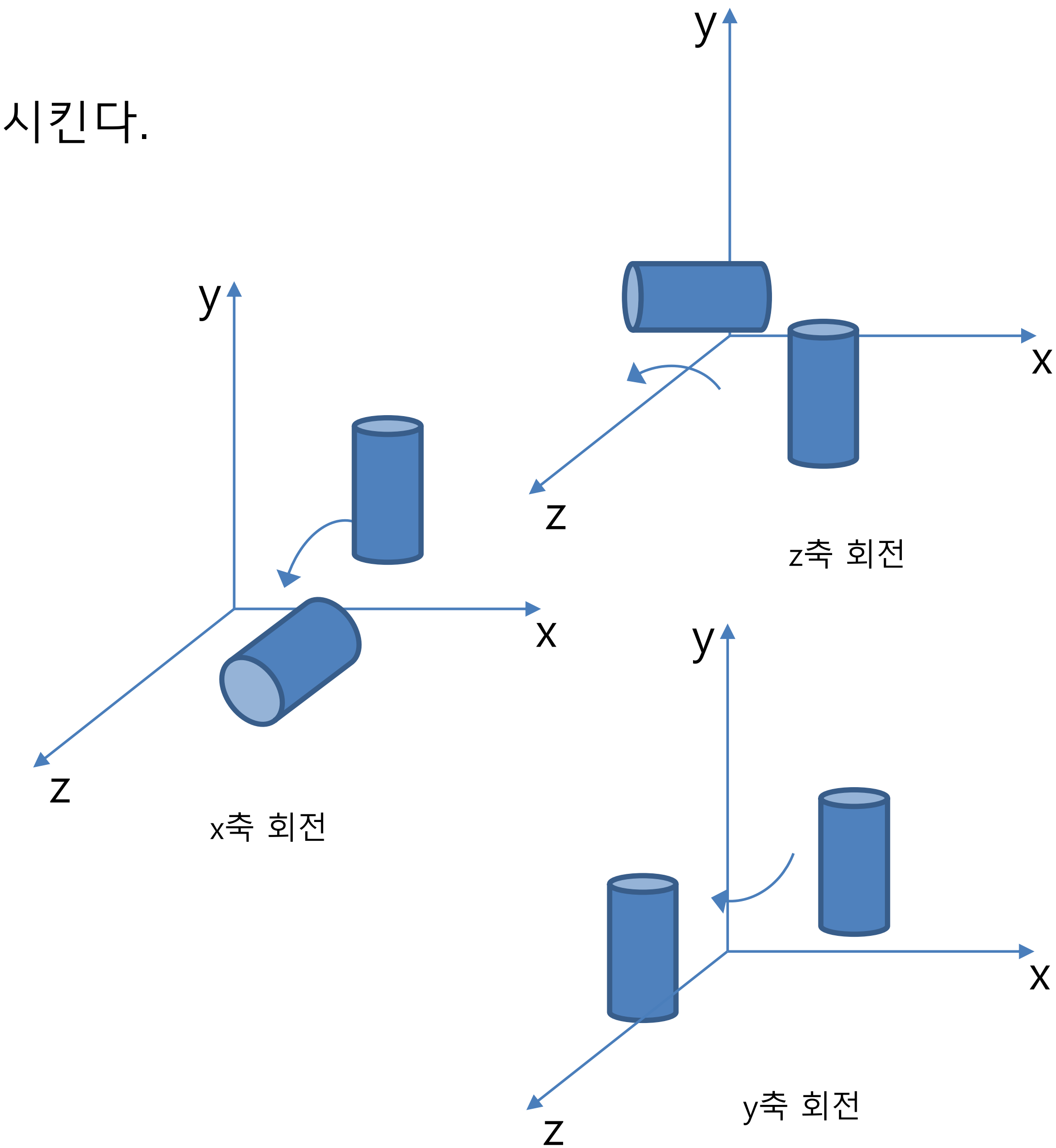
    $y' = y\cos\theta - z\sin\theta$

    $z' = y\sin\theta + z\mathrm{con}\,\theta$
  - y-축 회전 : $P' = R_y(\theta) \bullet P$

    $x' = x\mathrm{con}\,\theta + z\sin\theta$

    $y' = y$

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

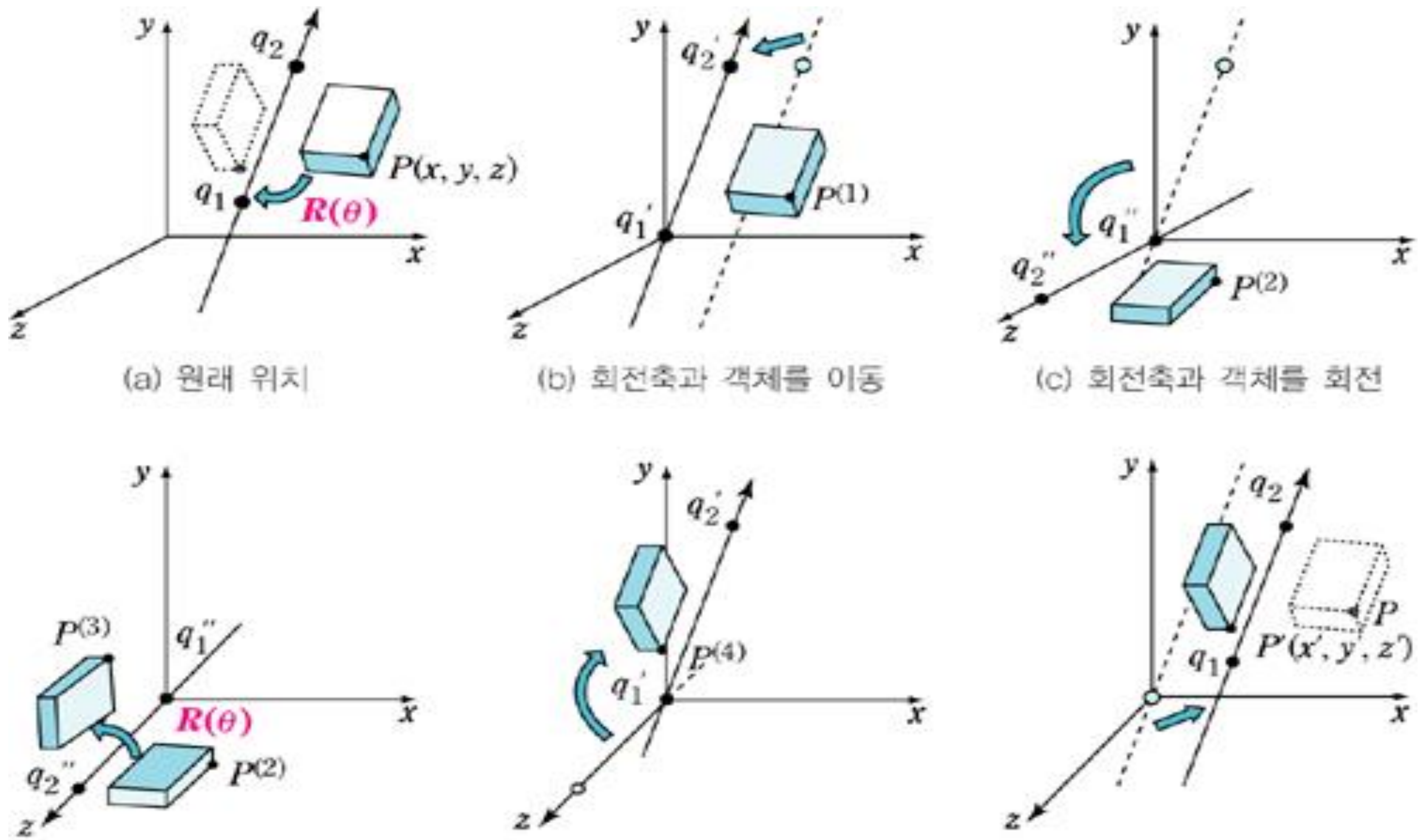


# 3차원 기하변환: 회전

- 임의의 축과 평행한 직선에 대한 회전
  - 회전축이 한 축이 되도록 이동
  - 그 축에 대해서 회전
  - 제자리로 역 이동

- 축과 평행하지 않은 일반적인 직선에 대한 회전
  - 회전축이 원점을 지나도록 이동시킨다.
  - 회전축을 좌표축 가운데 하나와 일치하도록 회전시킨다.
  - 일치된 좌표축을 중심으로 회전시킨다.
  - 2단계의 반대방향으로 회전한다.
  - 1단계의 반대방향으로 이동시킨다.

# 3차원 기하변환: 회전



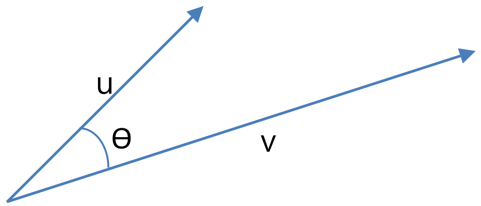

(a) 원래 위치 (b) 회전축과 객체를 이동 (c) 회전축과 객체를 회전

(d) **z**축 주위로 객체를 회전 (e) 반대 방향으로 회전 (f) 반대 방향으로 이동

$$M = T(P_0)R_x(-\Theta_x)R_y(-\Theta_y)R_z(\Theta)\,R_y(\Theta_y)\,R_x(-\Theta_x)\,T(-P_0)$$

# 간단한 수학

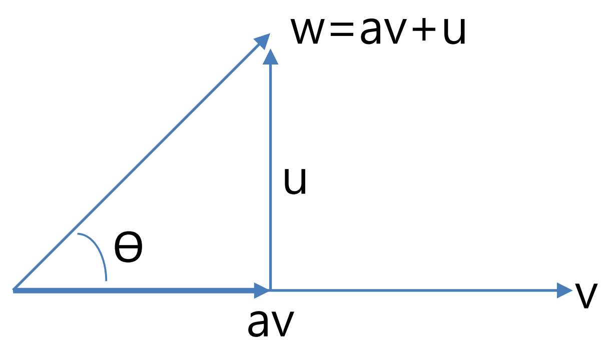
- **벡터 내적**
  - 두 벡터 사이의 각에 대한 코사인을 두 벡터의 길이 곱으로 스케일 한 스칼라 값
    - u ·v = ||u|| · ||v|| · cosΘ




  - 수학적으로 두 벡터 v1과 v2의 내적:
    - v1 · v2 = v1.x · v2.x + v1.y · v2.y + v1.z · v2.z

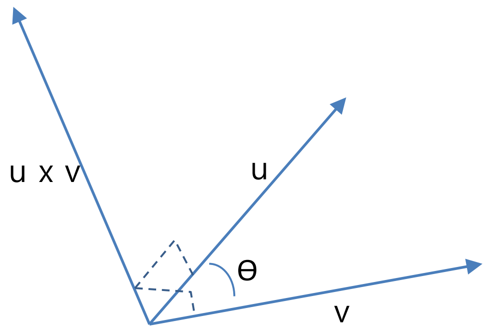
# 간단한 수학

- 두 개의 벡터 w, v가 주어지면, 그 중 하나의 벡터 w를 다른 벡터 v에 평행한 성분과 직교하는 성분으로 나눌 수 있다.
  - $w = w_x + w_y$
  - $w = av + u$
  - u는 v에 직교해야 하므로, $u \cdot v = 0$
  - $w \cdot v = (av+u) \cdot v = av \cdot v + u \cdot v = av \cdot v$
  - $a = (w \cdot v) / (v \cdot v)$
  - $u = w - av = w - \{(w \cdot v)/(v \cdot v)\} \cdot v = w - \{(w \cdot v)/||v||^2\} \cdot v$
  - $av = w - u = w - [w - \{(w \cdot v)/||v||^2\} \cdot v\ ] = \{(w \cdot v)/||v||^2\} \cdot v$



# 간단한 수학

- **벡터의 외적**
  - 처음 두 벡터가 놓인 평면에 수직인 세 번째 벡터
    - u x v = ||u|| ||v|| sinθ n (n: u와 v에 수직인 단위 벡터)
      - 두 벡터로 생성된 평행사변형의 영역

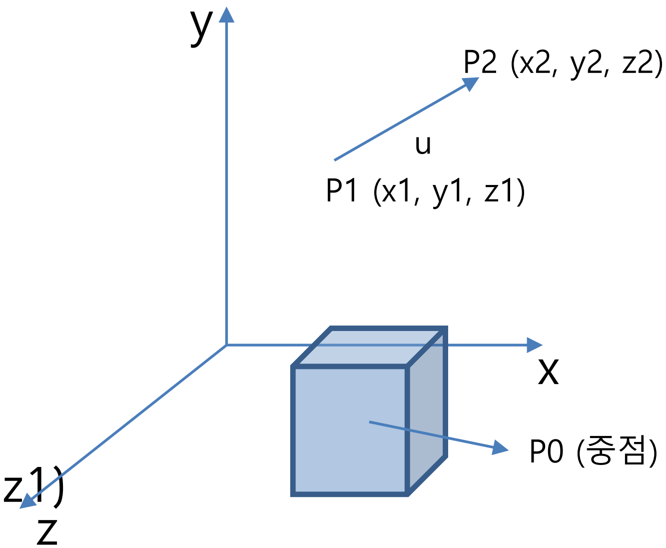



  - 수학적으로 두 벡터 v1과 v2의 외적:
    - v1 x v2 = (x1, y1, z1) x (x2, y2, z2) = (y1z2 – z1y2, z1x2 – x1z2, x1y2 – y1x2)

# 3차원 기하변환: 회전

- **임의의 점에 대하여 회전하기 위하여**
  - 원점을 지나가도록 이동: T(-P0)

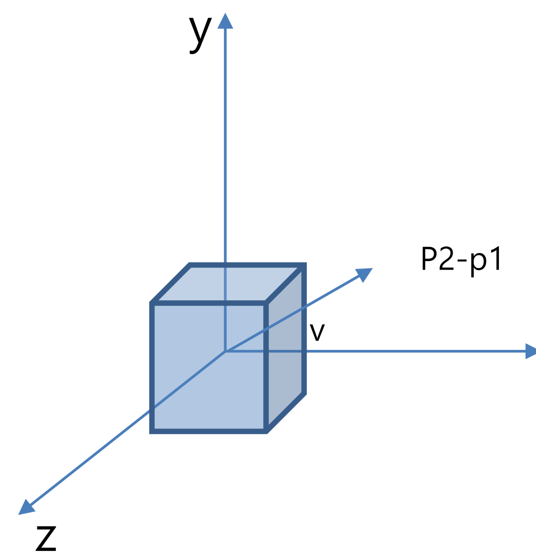
$$T = \begin{bmatrix} 1 & 0 & 0 & -x0 \\ 0 & 1 & 0 & -y0 \\ 0 & 0 & 1 & -z0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$$




  - 회전축 벡터: u = P2 - P1 = (x2 - x1, y2 - y1, z2 - z1)
  - u를 정규화 하기

$$v = \frac{u}{|u|} = \begin{bmatrix} a_x \\ a_y \\ a_z \end{bmatrix}$$

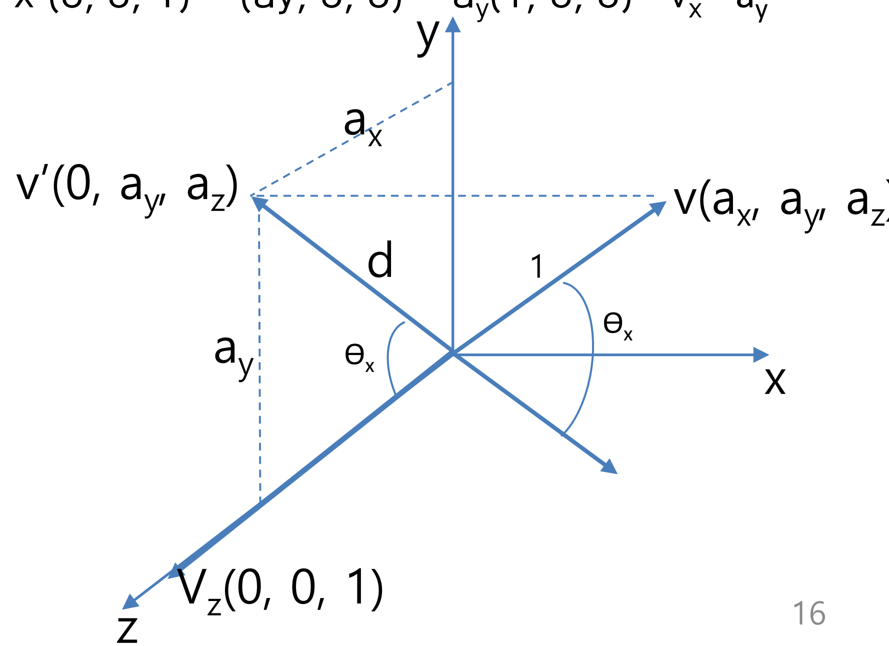
    - $v = (a_x, a_y, a_z)$, $a_x^2 + a_y^2 + a_z^2 = 1$




  - x축에 대하여 회전: 벡터는 xz 평면에 놓인다.
  - y축에 대하여 회전: 벡터는 z축에 놓인다.

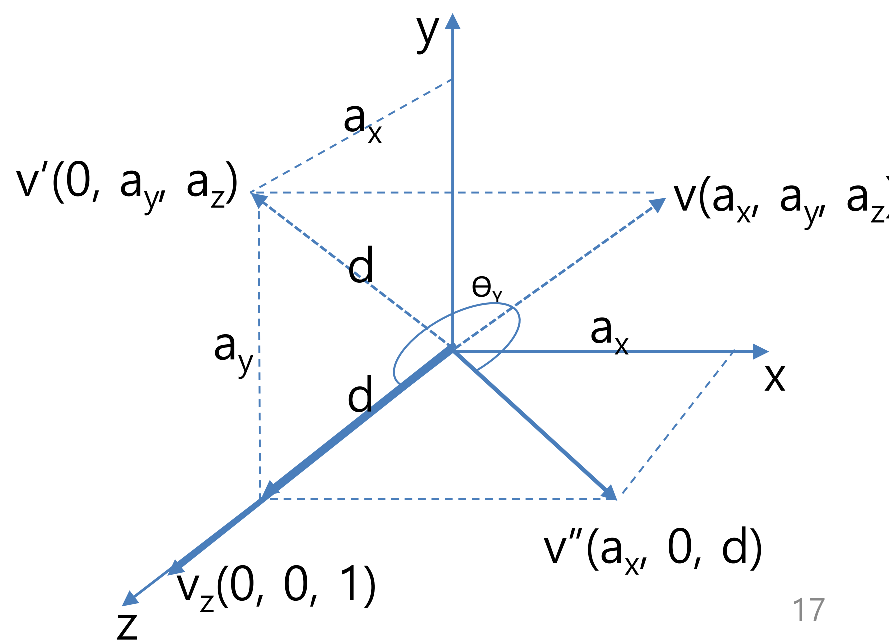
# 3차원 기하변환: 회전

- **회전축이 xz평면에 놓이도록 x축 회전**
  - $v'$와 z축의 단위 벡터 $u_z$ 사이의 내적: $v' \cdot v_z = |v'||v_z| \cos\theta_x$
    - $|v'| = \sqrt{a_y^2 + a_z^2}$ 이고 이 값을 d라고 하면,
    - $v_z = (0, 0, 1)$이고 $|v_z| = 1$
    - 수학적으로, $v' \cdot v_z = (0, ay, az) \cdot (0, 0, 1) = a_z$
    - 따라서, $\cos\theta_x = v' / |v'| = a_z / d$
  - $v'$와 z축의 단위 벡터 $u_z$ 사이의 외적: $v' \times v_z = |v'||v_z|\, v_x \sin\theta_x$
    - 직교좌표 형식으로, $v' \times v_z = (0, ay, az) \times (0, 0, 1) = (ay, 0, 0) = a_y(1, 0, 0) = v_x \cdot a_y$
    - 따라서, $\sin\theta_x = a_y / d$
  - 따라서, x축에 대한 $\theta_x$의 회전 행렬은
  - $R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \frac{a_z}{d} & -\frac{a_y}{d} & 0 \\ 0 & \frac{a_y}{d} & \frac{a_z}{d} & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



# 3차원 기하변환: 회전

- **회전축이 z축이 되도록 y축 회전**
  - v''와 z축의 단위 벡터 $u_z$ 사이의 내적: $v'' \cdot v_z = |v''||v_z| \cos\theta_y$
    - $|v_z| = 1$, $|v''| = \sqrt{ax^2 + ay^2 + az^2} = 1$
    - 수학적으로, $v'' \cdot v_z = (ax, 0, d) \cdot (0, 0, 1) = d$ ( $d = \sqrt{a_y^2 + a_z^2}$ )
    - 따라서, $\cos\theta_y = d$
  - v''와 z축의 단위 벡터 $u_z$ 사이의 외적: $v'' \times v_z = |v''||v_z| \, v_y \sin\theta_y$
    - 직교좌표 형식으로 $v'' \times v_z = (a_x, 0, d) \times (0, 0, 1) = (0, -a_x, 0) = -a_x(0, 1, 0)$
    - 따라서, $\sin\theta_y = -a_x$
  - 따라서, y축에 대한 $\theta_y$의 회전 행렬은
  - $R_y(\theta_y) = \begin{bmatrix} d & 0 & -a_x & 0 \\ 0 & 1 & 0 & 0 \\ a_x & 0 & d & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$



# 3차원 기하변환: 회전

- **Z축 회전**
  - $R_z\ (\theta_z) = \begin{bmatrix} \cos\theta_z & -\sin\theta_z & 0 & 0 \\ \sin\theta_z & \cos\theta_z & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

- **회전 후 제자리로 가기 위하여 역회전과 역 이동**
  - 역회전 $R_x\ (\theta_x)$, $R_y\ (-\theta_y)$
  - 역이동 T(P0)

- 최종적으로,

$$M = T(P_0)R_x(-\theta_x)R_y(-\theta_y)R_z(\theta)\ R_y(\theta_y)\ R_x(\theta_x)\ T(-P_0)$$

# 기타 3차원 기하변환: 반사

- Reflection (반사)
  - xy 평면에 반사
    - Z축 값의 부호가 바뀐다.

      $x' = x,\ \ y' = y,\ \ z' = -z$

  - yz 평면에 반사
    - X축 값의 부호가 바뀐다.

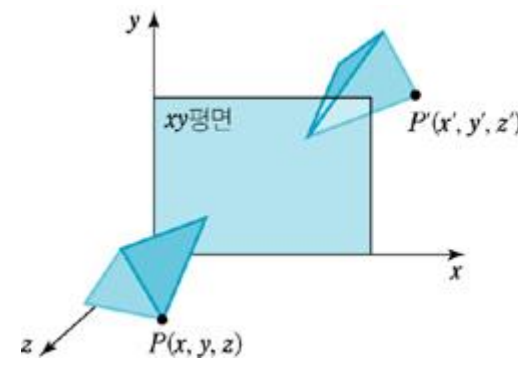
      $x' = -x, y' = y, z' = z$

  - xz 평면에 반사
    - Y축 값의 부호가 바뀐다.

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

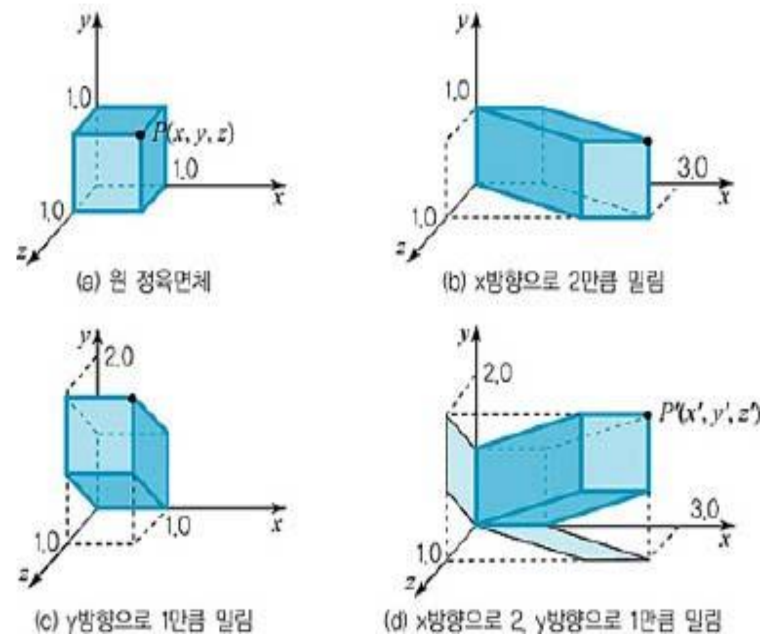
  - 원점에 반사
    - 모든 좌표값의 부호가 바뀐다.

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



# 기타 3차원 기하변환: 밀림

- Shearing (밀림)
  - x축을 기준으로 밀림 변환
    - x′ = x
    - y′ = y + ax
    - z′ = z + bx
      - a, b: 각각 y축과 z축 방향으로 밀리는 정도
      - x축을 기준으로 한 밀림이므로 x값은 변함이 없다.
  - y축을 기준으로 밀림 변환
    - x′ = x + ay
    - y′ = y
    - z′ = z + by
      - a, b: 각각 x축과 z축 방향으로 밀리는 정도
      - y축을 기준으로 한 밀림이므로 y값은 변함이 없다.
  - z축을 기준으로 밀림 변환
    - x’ = x + az
    - y’ = y + bz
    - z’ = z
      - a, b: 각각 x축과 y축 방향으로 밀리는 정도
      - z축을 기준으로 한 밀림이므로 z값은 변함이 없다.

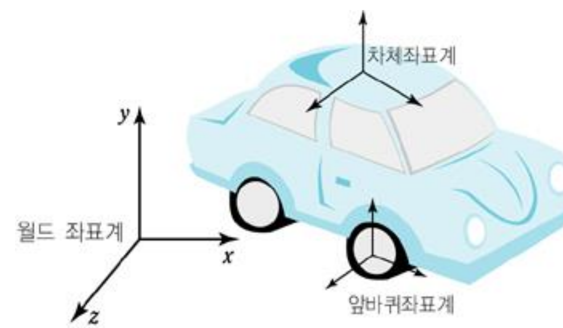



Z축을 기준으로 하는 밀림 변환

# 좌표계의 변환

- 객체를 고정시키고 좌표계를 변환시켜도 객체를 변환시킨 효과
  - 반대 방향으로 이동 또는 회전 시킨 효과
  - 좌표축을 확대/축소 하면 객체는 축소/확대 되는 효과
  - 여러 개의 객체를 묶어서 새로운 객체를 만드는 경우, 한번에 처리 가능
  - 뷰잉 과정에서 이용되며, 애니메이션 효과

- 자동차를 모델링한 예
  - 앞 바퀴를 표현하기 위한 좌표계와 차체를 표현하기 위한 좌표계가 상이
  - 자동차를 표현하기 위해서는 하나의 통합된 좌표계가 필요
  - 통합 좌표계와 앞 바퀴 좌표계간, 통합 좌표계와 차체 좌표계간에는 좌표변환이 필요



# 투영 (Projection)

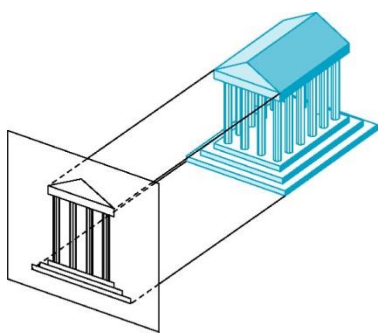
- **투영**
  - 3차원 공간상의 그래픽 개체를 2차원 평면에 표현하여 그래픽 화면을 만들어 내는 과정

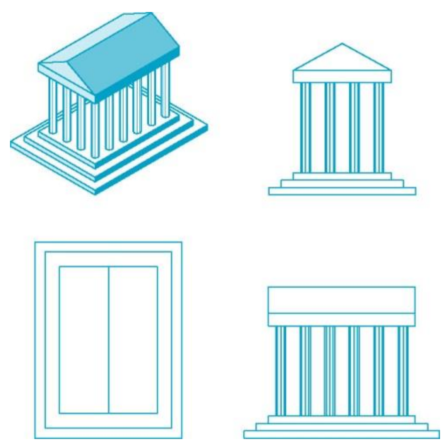
  - 평행 투영(Parallel Projection)
    - 출력면에 수평선의 선을 따라 물체 표면의 점들을 투영하는 방법
    - 객체들간의 상대적인 크기 정보가 보존된다.
    - 다른 view에 따라 물체의 다른 2차원 view를 얻을 수 있다.

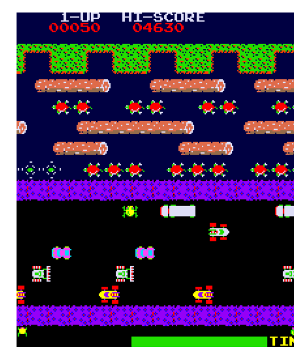
  - 원근 투영 (Perspective Projection)
    - 공간상의 객체와 투영 중심점 (view point)를 연결하여 투영
    - 투영면과 시점이 먼 객체는 작게, 가까운 객체는 크게
    - 현실적인 결과

# 투영: 평행 투영

- **평행 투영 (Parallel Projection)**
  - 직각 투영 (Orthographic Projection)
    - 투영방향과 투영면이 직각을 이루는 경우
    - 임의의 점 P(x, y, z) → P'($x_p$ , $y_p$ , $z_p$)
      - $x_p = x$  $y_p = y$  $z_p = 0$
      - Front view (z 값 삭제): 입면도, 정면도
      - Side view (x 값 삭제): 측면도
      - Rear view (z 값 삭제)
      - Top view (y 값 삭제): 평면도
      - 엔지니어링, 건축에서 많이 사용한다 (길이와 각도가 정확하다)

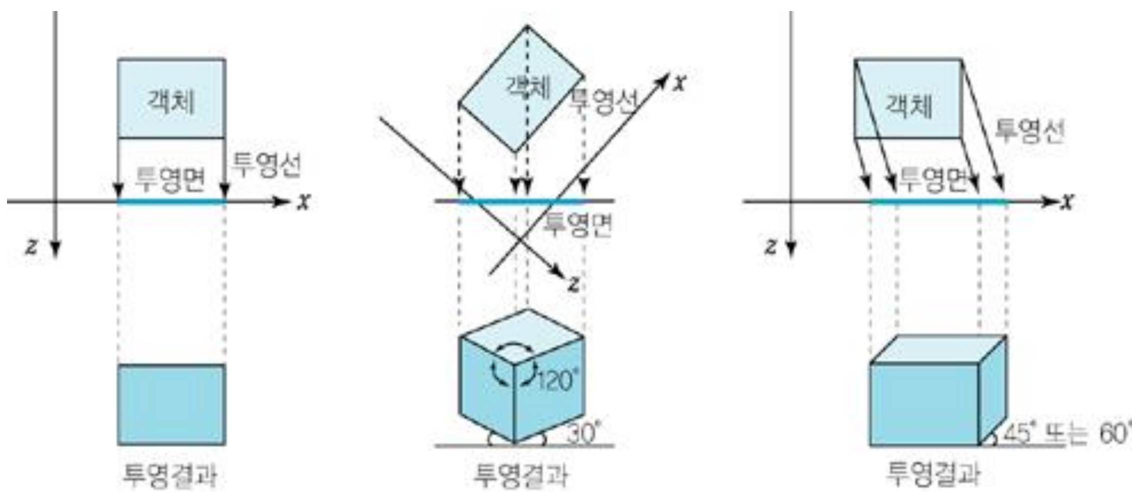






# 투영: 평행 투영

- 경사 투영(Oblique Projection)
  - 객체의 투영방향이 투영면과 수직이 아닌 일정한 각도를 이루는 경우
  - <u>2개의 각도로 정의</u>
    - **각도 α (투영 각도):** 점 (x, y, z)과 경사투영의 점 ($x_p$, $y_p$)의 선, 점 (x, y, z)과 직각투영의 점 (x, y)의 선이 만드는 각도
    - **각도 ϕ:** 점 (x, y)와 점 ($x_p$, $y_p$)의 선과 투영면에 평행한 방향과의 각도



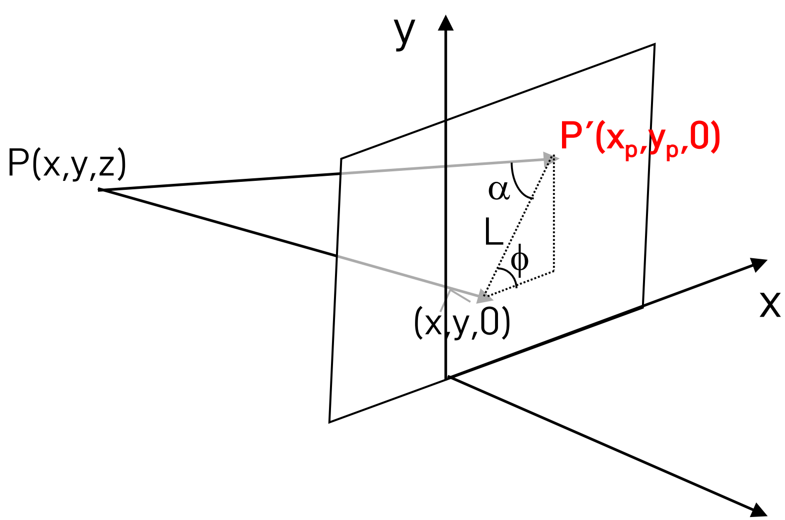
(a) 직각투영 (b) 등축투영 (c) 경사투영





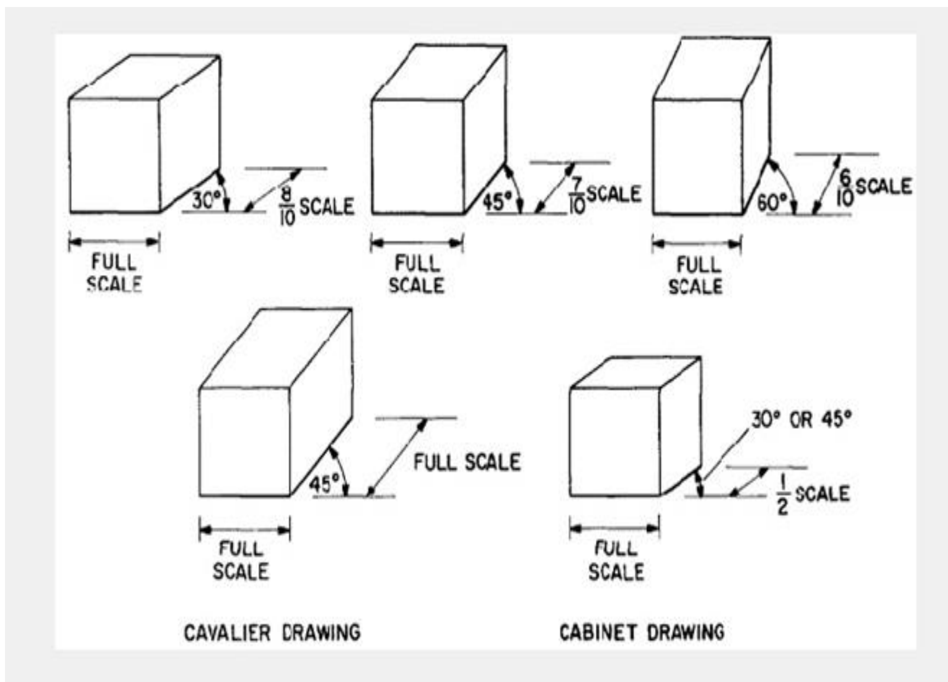
# 투영: 평행 투영

- 경사 투영에서
  - 투영면: $z = 0$
  - 공간상의 점:        $P\ (x, y, z)$
  - 경사 투영된 점: $P'\ (x_p, y_p, z_p)$
    투영면이 $z=0$이므로 $P' = (x_p, y_p, 0)$
  - 투영선과 투영면의 각도: $\alpha$
  - 점P가 직각 투영된 점과 경사 투영된 점을 연결한 선분의 길이: L
  - L과 x축과 이루는 각도: $\phi$
    - $\cos\phi = (x_p - x) / L$ $\rightarrow x_p = x + L\cos\phi$
    - $\sin\phi = (y_p - y) / L$ $\rightarrow y_p = y + L\sin\phi$
    - $\tan\alpha = z / L$ $\rightarrow L = z / \tan\alpha = zL_1$
  - $x_p = x + L\cos\phi = x + z(\cos\phi / \tan\alpha)$
  - $y_p = y + L\sin\phi = y + z(\sin\phi / \tan\alpha)$



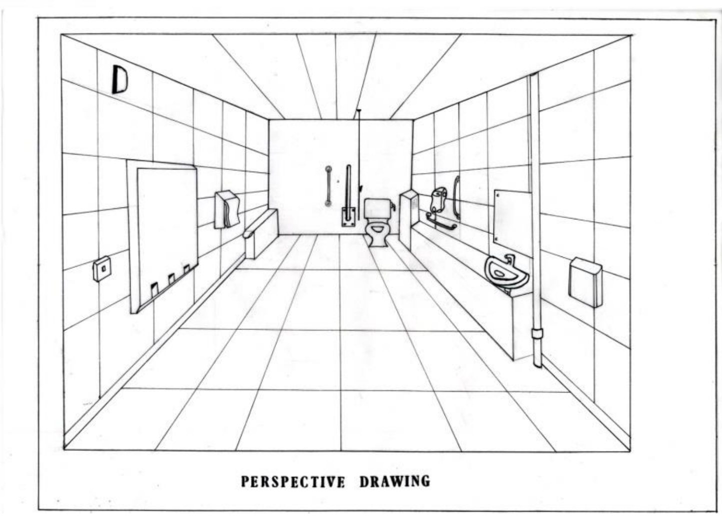
# 투영: 평행 투영

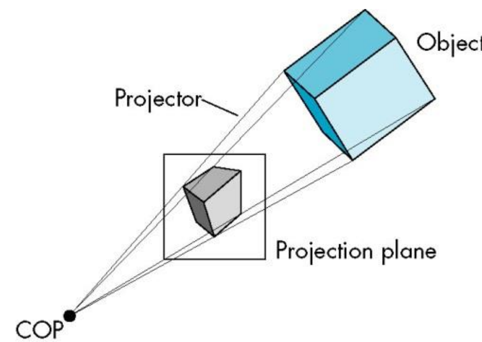
- 투영 각도 α에 대해서
  - α = 45’ (tan α = 1) 인 경우: cavalier  투영
    - 투영면에 수직인 선들은 길이 변환이 없고, 정육면체의 깊이는 폭과 높이가 같은 길이로 투영된다.
  - α = 63.4’ (tan α = 2)인 경우: cabinet 투영
    - 투영면과 수직인 선들은 그들 길이의 절반으로 투영되고 깊이가 폭과 높이의 절반으로 투영된다.

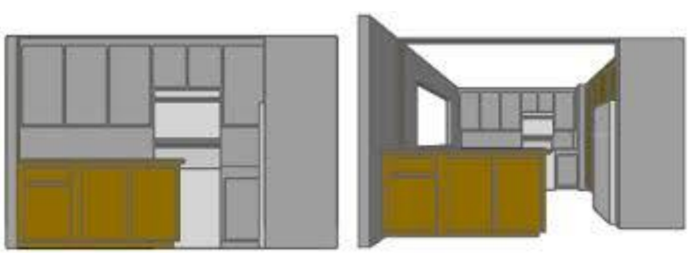


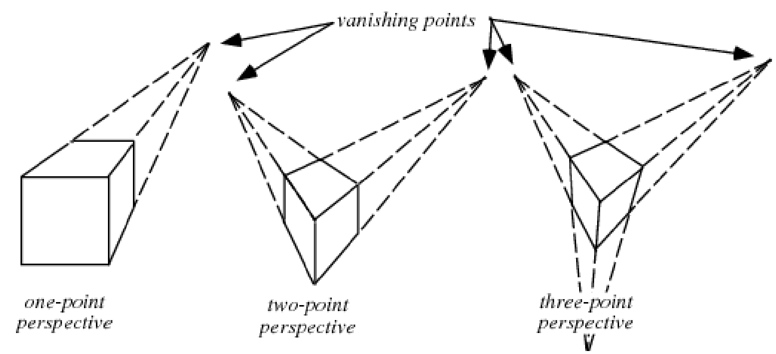
# 투영: 원근 투영

- Perspective Projection (원근 투영)
  - 객체와 투영중심점 (시점, view point)을 연결하여 투영 면에 2차원 객체를 만든다.
  - 투영면에서 멀리 떨어진 객체는 작게, 가까운 객체는 크게 나타나 현실감 있는 결과를 얻는다.



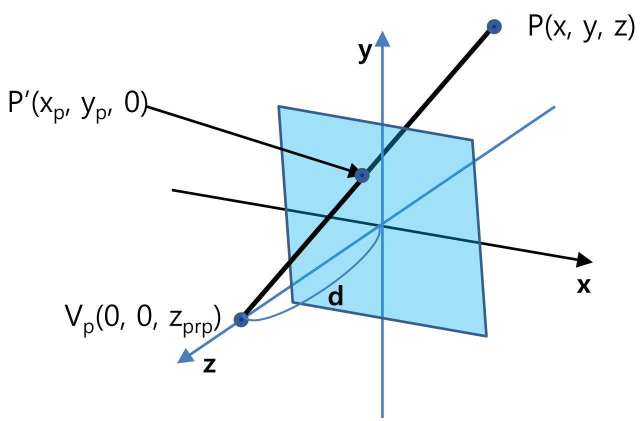


# 투영: 원근 투영

- Z축 위의 임의의 점 로 투영할 때
  - 투영 참조점: $z_{prp}$ 투영 면: $z_{vp}$

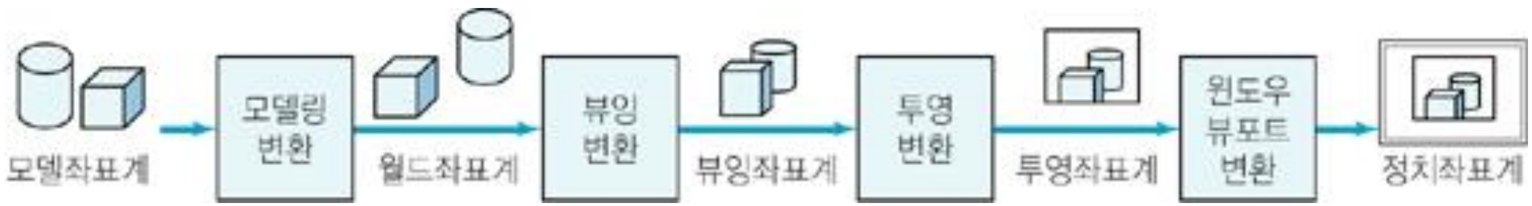
- 점 P(x, y, z)을 z축에 따라 투영면 (z = 0)에 원근 투영시키면,
  - 투영점 $P'(x_p, y_p, z_{vp})$, 투영참조점 좌표를 $(0, 0, z_{prp})$라 하면
    - $u = (z - z_{vp}) / (z - z_{prp}) = |z| / (|z| + d)$
      - |z|: (x, y, z)에서 투영면까지의 거리
      - d: 투영면에서 투영 참조점까지의 거리
  - 매개 변수 u : $0 \leq u \leq 1$ 의 값으로
    - $u = 0 \rightarrow u = |z|/(|z|+d) = 0 \rightarrow |z| = 0 \rightarrow P' = (x, y, z)$
    - $u = 1 \rightarrow u = |z|/(|z|+d) = 1 \rightarrow d = 0 \rightarrow P' = (0, 0, z_{prp})$
  - 매개변수 u를 사용하여
    - $x_p = (1-u)x_1 + ux_2 = x_1 - x_1u = x - x\,(z/(z+d))$
      ($x_1$ 은 x, $x_2$ 는 0)
    - $y_p = (1-u)y_1 + uy_2 = y_1 - y_1u = y - y\,(z/(z+d))$
      ($y_1$ 은 y, $y_2$ 는 0)

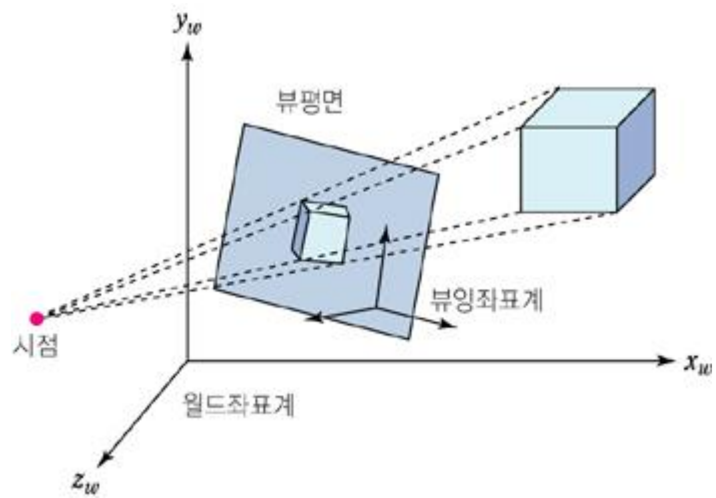
    행렬로 나타내면,



# 뷰잉 변환

- **뷰잉 과정**
  - 3차원 객체들을 하나의 좌표계로 통합한 후 투영되어 출력 화면에 나타나게 되는 과정
  - 뷰잉 변환

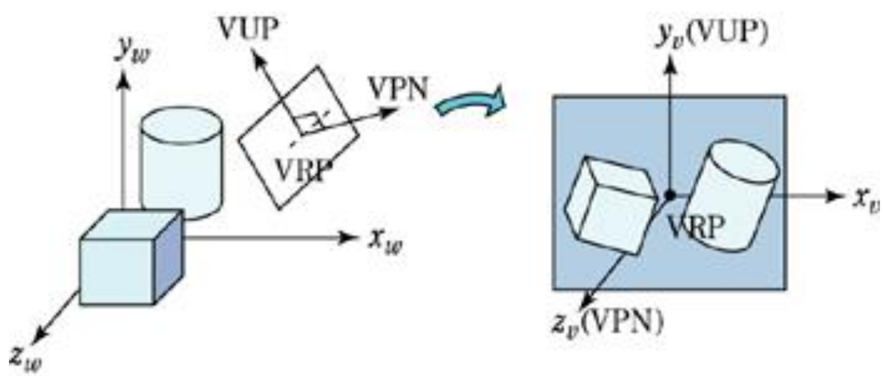


# 뷰잉 변환

- 투영 과정을 용이하게 처리하기 위해 월드 좌표계를 뷰잉 좌표계로 변환
  - 투영면이 z = 0 인 xy 평면으로 된다

- 뷰 평면의 축 벡터와 법선 벡터를 이용하여 설정
  - 원점: 뷰 평면 상의 한 점 (카메라 위치)
  - Normal Vector: z축에 해당 (바라보는 방향)
  - Up Vector: y축에 해당 (x축은 자동으로 결정) (카메라 각도)

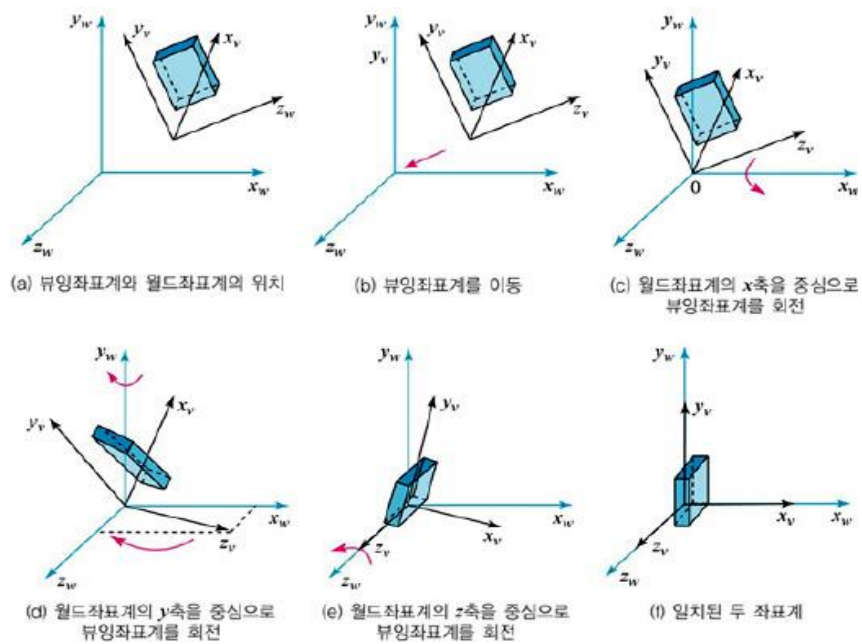



(a) 월드좌표계와 뷰평면 (b) 뷰평면에 나타난 객체

VRP: View Reference Point
VPN: View Plane Normal Vector
VUP: View Up Vector

# 좌표계 변환

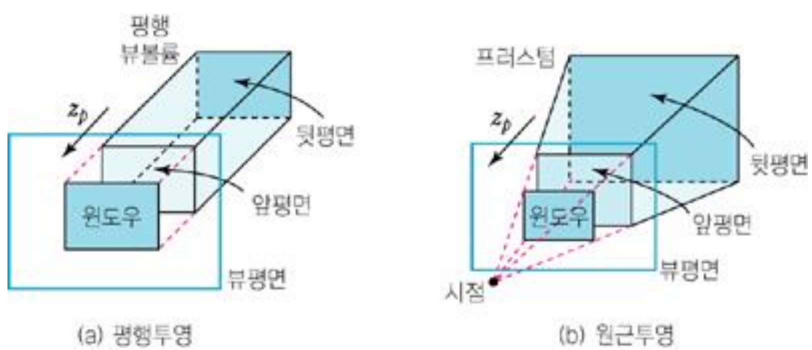
- World Coordinate → Viewing Coordinate
  - 뷰잉좌표계가 주어짐
  - 뷰잉좌표계 원점을 월드좌표계 원점과 일치하도록 이동
  - 월드좌표계의 X축을 중심으로 뷰잉좌표계의 Z축을 회전
    - 뷰잉좌표계의 z축이 월드좌표계의 zx 평면에 위치
  - 월드좌표계의 Y 축을 중심으로 뷰잉좌표계를 회전- 두 좌표계의 z축이 일치
  - 월드좌표계의 Z축을 중심으로 뷰잉좌표계를 회전- 뷰잉좌표계와 월드 좌표계가 일치




(a) 뷰잉좌표계와 월드좌표계의 위치
(b) 뷰잉좌표계를 이동
(c) 월드좌표계의 $x$축을 중심으로 뷰잉좌표계를 회전
(d) 월드좌표계의 $y$축을 중심으로 뷰잉좌표계를 회전
(e) 월드좌표계의 $z$축을 중심으로 뷰잉좌표계를 회전
(f) 일치된 두 좌표계

# 투영을 위한 변환

- **뷰평면의 윈도우 내에 투영되는 공간상의 일정영역**
  - 투영 변환에서 뷰평면의 윈도우에 투영되는 객체들은 3차원 공간에서 일정한 영역 내에 존재: 뷰볼륨
    - 평행 투영의 경우: 평행 뷰볼륨
    - 원근 투영의 경우: 프러스텀(Frustum) 뷰볼륨
  - 뷰볼륨을 직육면체 형태로 변환하여 직각투영을 이용하면, 투영과 클리핑이 간단해진다
  - 정규화된 뷰볼륨
    - 모든 좌표를 0과 1사이의 값으로 표현, 정육면체 형태
    - 장치 좌표계로의 변환 용이, 클리핑 과정이 매우 단순화



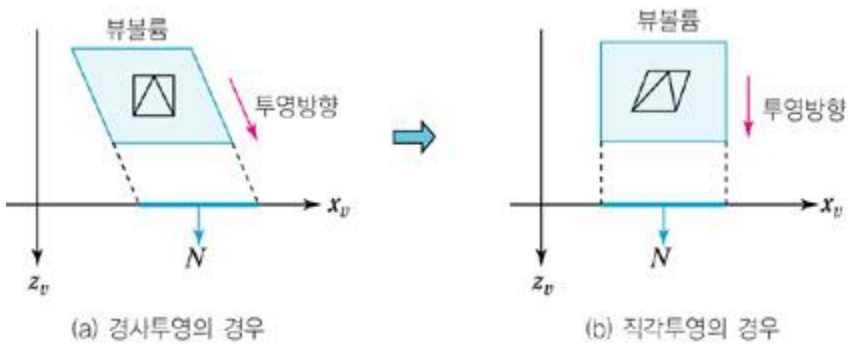
# 투영을 위한 변환

- **평행 투영의 변환 행렬**
  - 직각 투영
    - 투영면이 xy평면(z=0)인 경우
    - 공간상의 점 P(x, y, z)가 직각 투영된 점은 (x, y, 0)이 된다 즉,

$$P' = \begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = M_{ortho} \cdot P$$

# 투영을 위한 변환

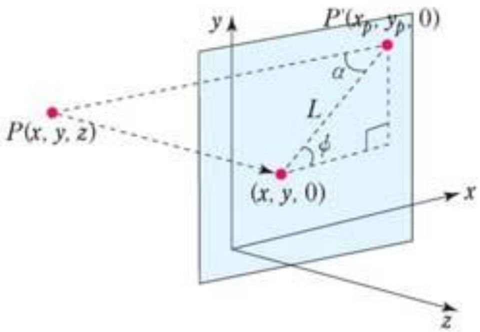
- **평행 투영의 변환 행렬**
  – 경사 투영
    - 기울어진 형태의 뷰볼륨을 직육면체 형태로 밀림 변환



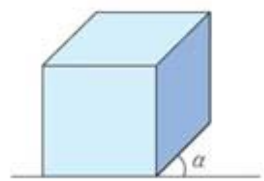
# 투영을 위한 변환

- 공간상의 점 P(x, y, z)가 경사 투영된 점 P´($x_p$, $y_p$, 0)을 구하려면

- 경사 각도 α와 투영길이 L로 정의
  - L: 경사 투영점과 직각 투영점간의 거리
  - ϕ : L과 x축과 이루는 각도
  - $\tan \alpha = z / L \quad \rightarrow \quad L = z / \tan \alpha \quad = z \cot \alpha$
  - $x_p = x + L\cos \alpha$
  - $y_p = y + L\sin \alpha$




(a) 공간상의 한 점이 경사투영된 경우

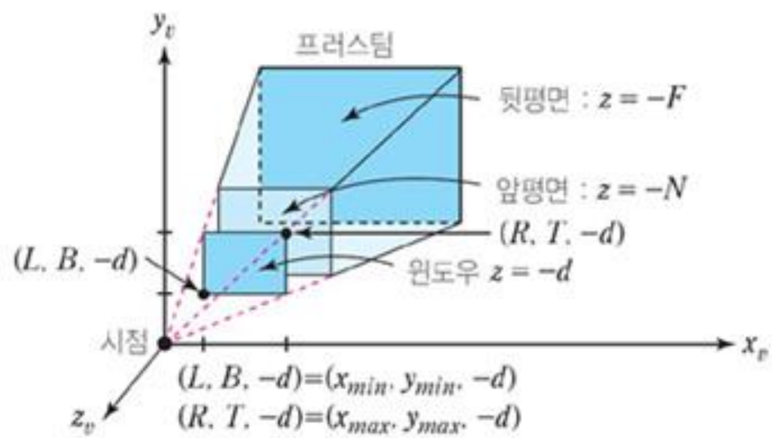


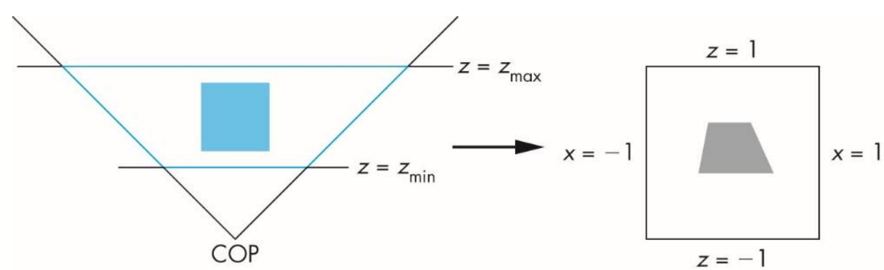

(b) 경사투영의 경우

$$P' = \begin{pmatrix} x_p \\ y_p \\ z_p \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \cot\alpha\cos\phi & 0 \\ 0 & 1 & \cot\alpha\sin\phi & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = M_{obliq} \cdot P$$

# 투영을 위한 변환

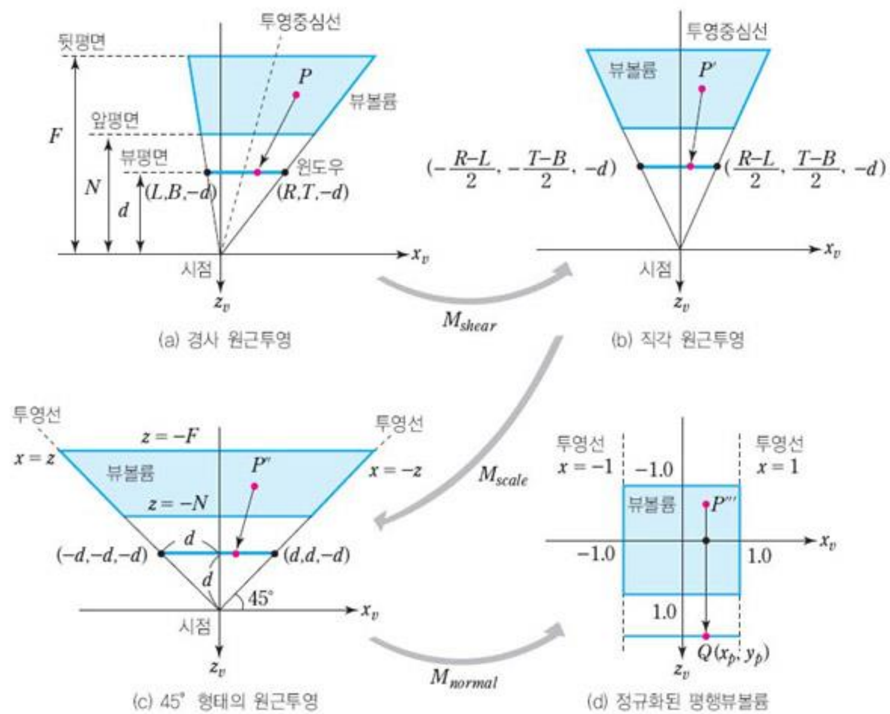
- **원근 투영의 변환 행렬**
  - 프러스텀을 직육면체 형태로 변환하여 직각투영 이용
    - 시점: 뷰잉 좌표계의 원점
    - 윈도우: 법선벡터는 z축 방향
    - 뷰평면 기준: left, right, top, bottom
      - d: 뷰 평면이 놓여진 z 값
    - 프러스텀  뒷 평면과 앞 평면: -F, -N



# 투영을 위한 변환

– 밀림변환과 신축변환을 수행
  - 과정 1: 경사원근투영을 직각원근투영의 뷰볼륨으로 변환
  - 과정 2: 직각원근투영의 뷰볼륨을 정육면체 형태로 변환
    – 45도 각도의 피라미드 형태의 뷰볼륨으로 변환
    – 피라미드 뷰볼륨을 정육면체 뷰볼륨으로 변환



# 투영을 위한 변환

- 과정 1:  밀림변환 적용 P 가 P' 으로 변환

$$P' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & \frac{R+L}{2d} & 0 \\ 0 & 1 & \frac{T+B}{2d} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = M_{shear} \cdot P$$

- 과정 2: 신축변환 적용 P′가 P′′로 변환

$$P'' = \begin{pmatrix} x'' \\ y'' \\ z'' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{2d}{R-L} & 0 & 0 & 0 \\ 0 & \frac{2d}{T-B} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = M_{scale} \cdot P'$$

- 과정 3: 정규화 적용,  P'' 이  P'''  으로 변환

$$P''' = \begin{pmatrix} x''' \\ y''' \\ z''' \\ h \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{F+N}{F-N} & -\frac{2FN}{F-N} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ z'' \\ 1 \end{pmatrix} = M_{normal} \cdot P''$$

- 따라서, 원근 투영 뷰볼륨의 전체 변환 과정은,

$$\begin{aligned} P''' &= M_{persp} \bullet P \\ &= M_{normal} \bullet M_{scale} \bullet M_{shear} \bullet P \end{aligned}$$