컴퓨터 그래픽스
7. 3차원 그래픽스의 투영 및 뷰잉

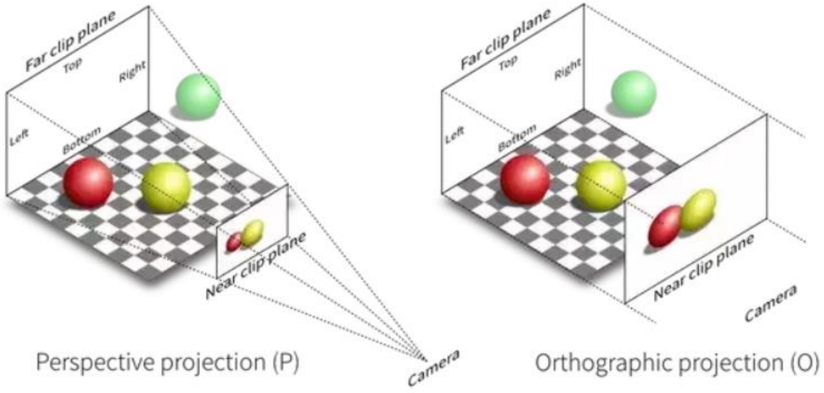
2025년 2학기

학습 내용

- 3차원 그래픽스의 투영 및 뷰잉
  - 투영
  - 뷰잉 변환

투영 (Projection)

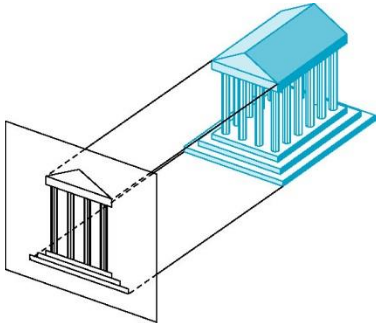
- **투영**
  - 3차원 공간상의 그래픽 개체를 2차원 평면에 표현하여 그래픽 화면을 만들어 내는 과정
  - 투영의 종류
    - 평행 투영(Parallel Projection)
      - 출력면에 수평선의 선을 따라 물체 표면의 점들을 투영하는 방법
      - 객체들간의 상대적인 크기 정보가 보존된다.
      - 다른 view에 따라 물체의 다른 2차원 view를 얻을 수 있다.
    - 원근 투영 (Perspective Projection)
      - 공간상의 객체와 투영 중심점 (view point)를 연결하여 투영
      - 투영면과 시점이 먼 객체는 작게, 가까운 객체는 크게
      - 현실적인 결과

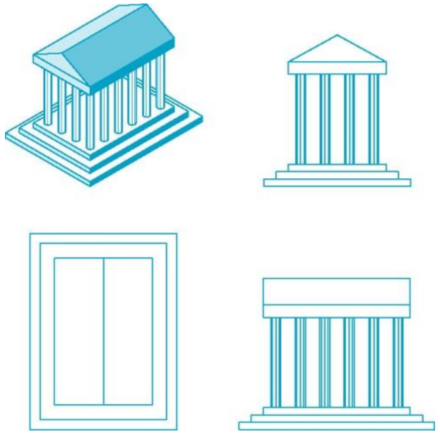


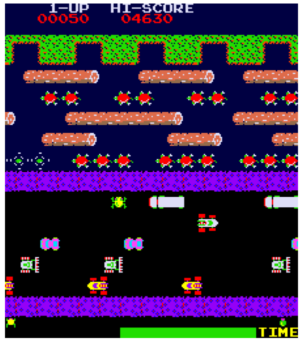
Perspective projection (P)
Orthographic projection (O)

투영: 평행 투영

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

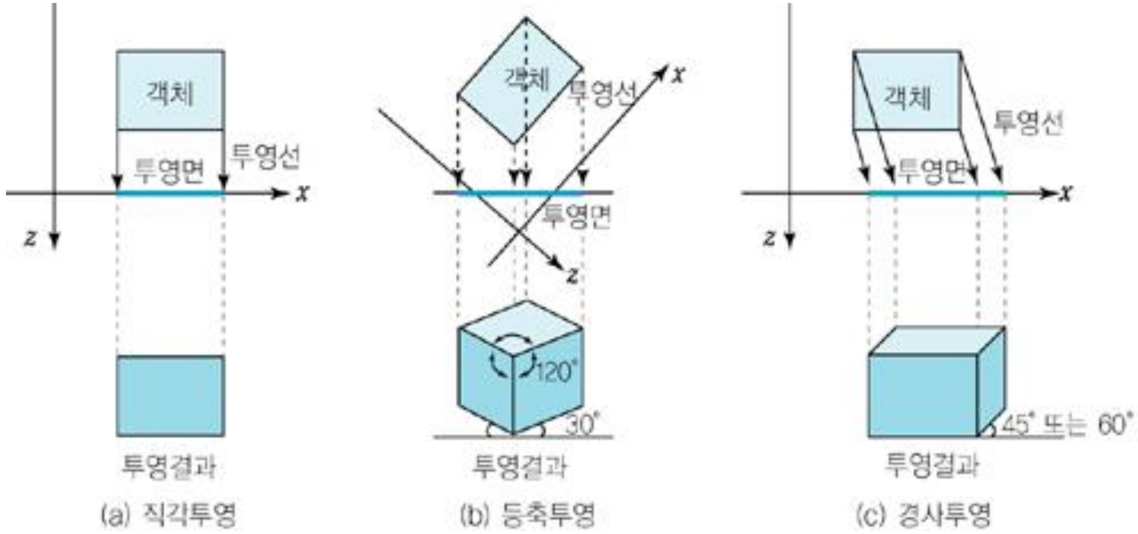






투영: 평행 투영

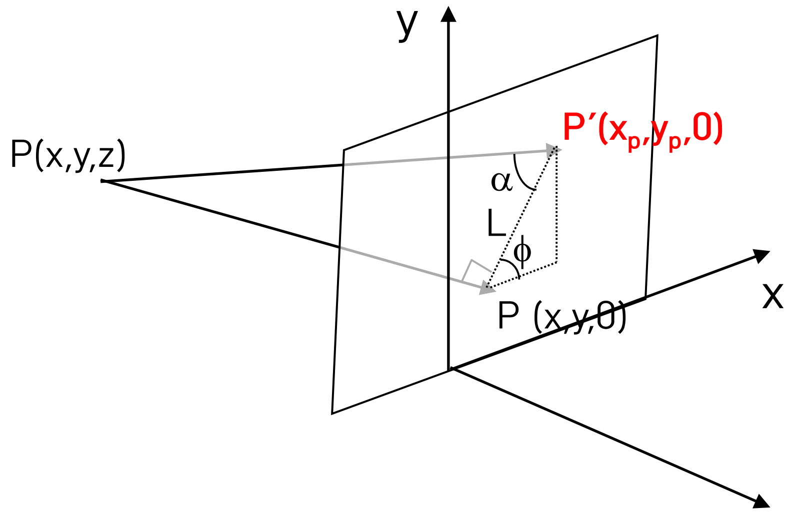
– 경사 투영(Oblique Projection)
  - 객체의 투영방향이 투영면과 수직이 아닌 일정한 각도를 이루는 경우
  - 2개의 각도로 정의
    - **각도 α (투영 각도):** 점 (x, y, z)과 경사투영의 점 (x_p, y_p)의 선, 점 (x, y, z)과 직각투영의 점 (x, y)의 선이 만드는 각도
    - **각도 ϕ:** 점 (x, y)와 점 (x_p, y_p)의 선과 투영면에 평행한 방향과의 각도



투영: 평행 투영

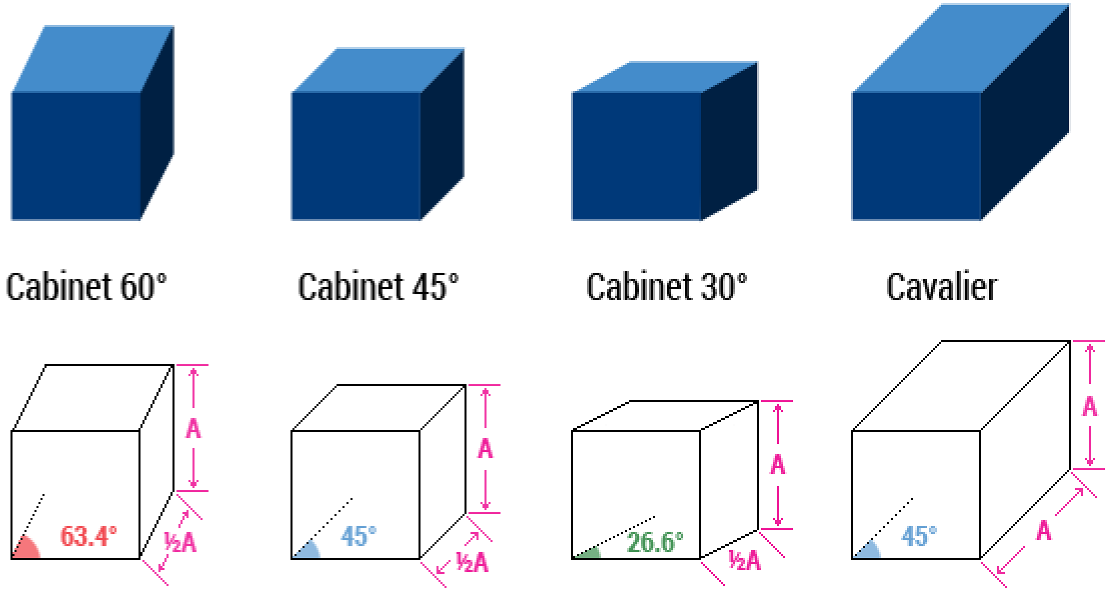
- 경사 투영에서
  - 투영면: z = 0인 xy 평면
  - 공간상의 점: P (x, y, z)
  - 경사 투영된 점: P' (x_p, y_p, z_p)
  - 투영면이 z=0이므로 P' = (x_p, y_p, 0)

  - 투영선과 투영면의 각도: α (투영 각도)
  - 점P가 직각 투영된 점과 경사 투영된 점을 연결한 선분의 길이: L
  - L과 x축과 이루는 각도: ϕ
    - $\cos\phi = \frac{(x_p - x)}{L}$ $\rightarrow x_p = x + L\cos\phi$
    - $\sin\phi = \frac{(y_p - y)}{L}$ $\rightarrow y_p = y + L\sin\phi$
    - $\tan\alpha = \frac{z}{L}$ $\rightarrow L = \frac{z}{\tan\alpha} = zL_1$
  - $x_p = x + L\cos\phi = x + z\frac{\cos\phi}{\tan\alpha}$
  - $y_p = y + L\sin\phi = y + z\frac{\sin\phi}{\tan\alpha}$



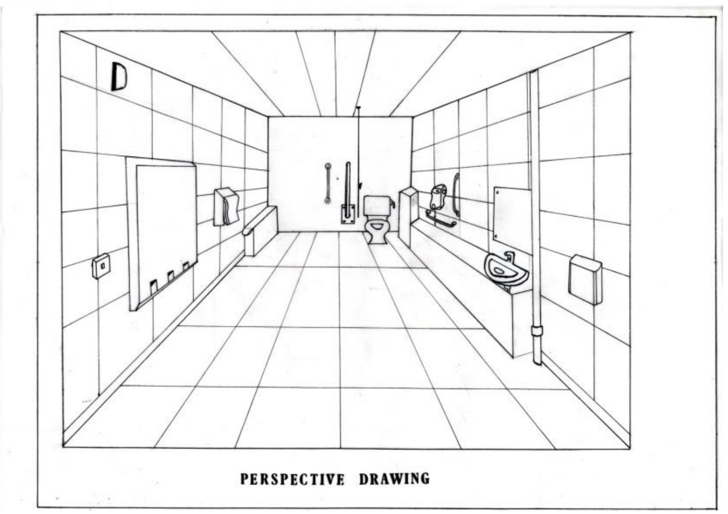
투영: 평행 투영

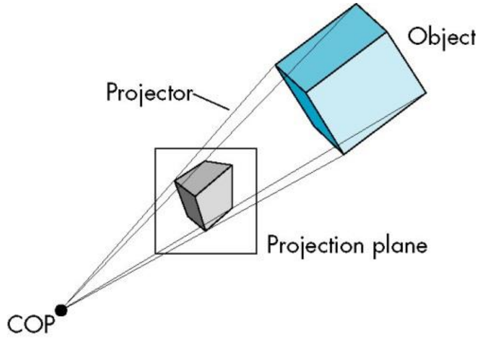
- 투영 각도 α에 대해서
  - α = 45’ (tan α = 1) 인 경우: cavalier 투영
    - 투영면에 수직인 선들은 길이 변환이 없고, 정육면체의 깊이는 폭과 높이가 같은 길이로 투영된다.
  - α = 63.4’ (tan α = 2)인 경우: cabinet 투영
    - 투영면과 수직인 선들은 그들 길이의 절반으로 투영되고 깊이가 폭과 높이의 절반으로 투영된다.

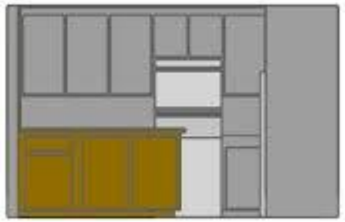


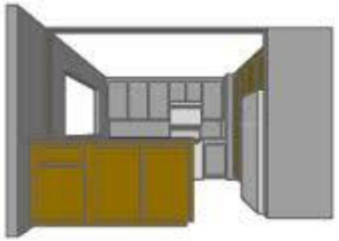
투영: 원근 투영

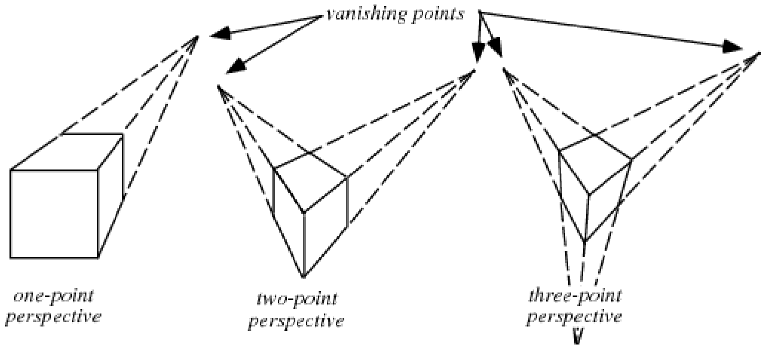
- Perspective Projection (원근 투영)
  - 객체와 투영중심점 (시점, view point)을 연결하여 투영 면에 2차원 객체를 만든다.
  - 투영면에서 멀리 떨어진 객체는 작게, 가까운 객체는 크게 나타나 현실감 있는 결과를 얻는다.









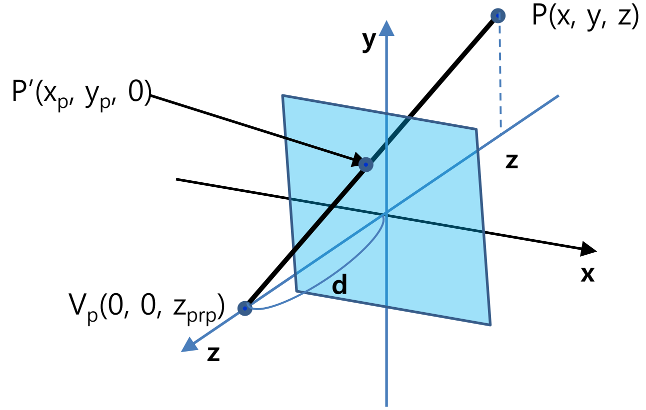


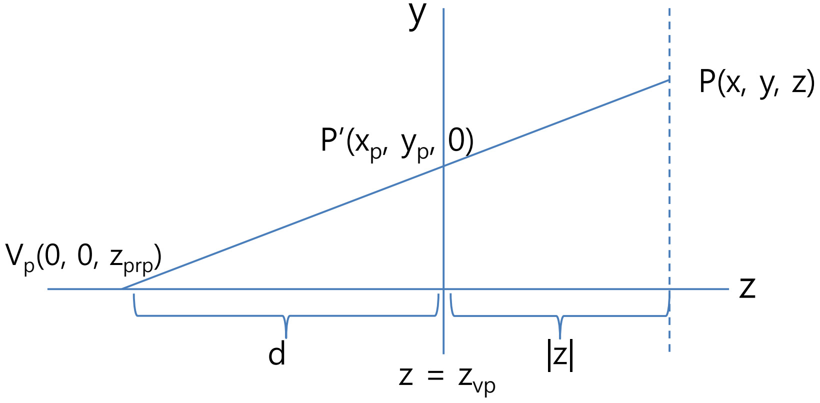
투영: 원근 투영

- 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: $(0, 0, z_{prp})$
    - $u = \dfrac{(z - z_{vp})}{(z - z_{prp})} = \dfrac{|z|}{|z| + d}$
      - |z|: (x, y, z)에서 투영면까지의 거리
      - d: 투영면에서 투영 참조점까지의 거리
  - 매개 변수 u : $0 \le u \le 1$ 의 값으로
    - $u = 0 \rightarrow u = \dfrac{|z|}{|z| + d} = 0 \rightarrow |z| = 0 \rightarrow P' = (x, y, z)$
    - $u = 1 \rightarrow u = \dfrac{|z|}{|z| + d} = 1 \rightarrow d = 0 \rightarrow P' = (0, 0, z_{prp})$
  - 매개변수 u를 사용하여
    - $x_p = (1-u)x_1 + ux_2 = x_1 - x_1u = x - x\dfrac{|z|}{|z| + d}$ $(x_1 = x, x_2 = 0)$
    - $y_p = (1-u)y_1 + uy_2 = y_1 - y_1u = y - y\dfrac{|z|}{|z| + d}$ $(y_1 = y, y_2 = 0)$

  행렬로 나타내면,





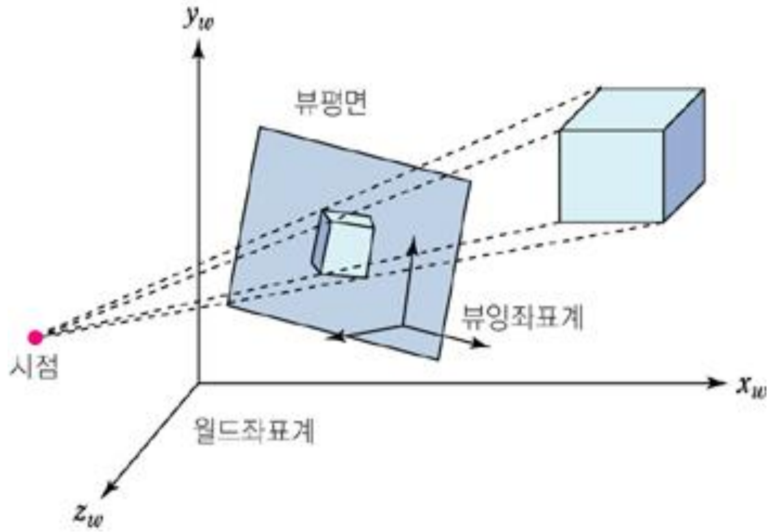
뷰잉 변환

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

  - 모델링 변환: 모델좌표계의 3D 객체들을 월드좌표계로 가져옴
  - 뷰잉 변환: 월드 좌표계에서 표현된 3D 객체들을 뷰잉 좌표계로 변환
  - 투영 변환: 뷰잉 좌표계로 변환된 3D 객체들을 2차원 뷰평면에 투영
  - 윈도우-뷰포트 변환: 투영 좌표계의 결과를 출력장치의 장치좌표계로 표현

뷰잉 변환

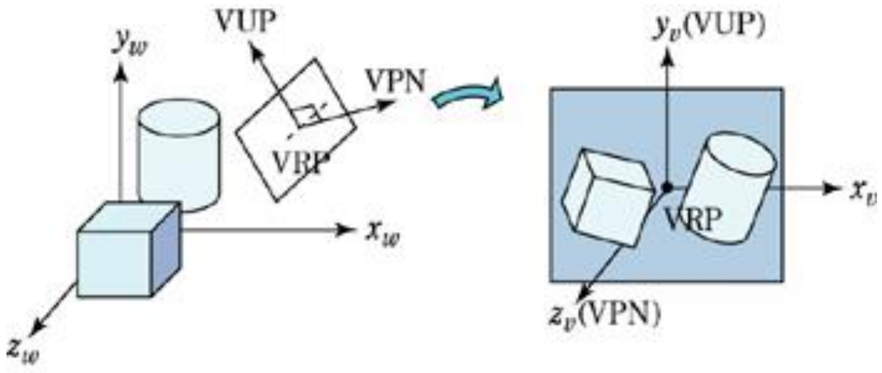
- **투영 과정을 용이하게 처리하기 위해 월드 좌표계를 뷰잉 좌표계로 변환**
  - 투영 방향은 z축 방향으로 한다.
  - 투영면은 z = 0 인 xy 평면으로 한다.



- **뷰잉 좌표계 설정**
  - 투영변환이 실행되기 위해서 지정해야할 요소들
    - 투영면 → 뷰 평면
    - 클리핑 공간 → 뷰 볼륨
    - 투영 종류에 따라 투영 방향 (투영 중심점)
  - 뷰잉 변환 단계에서 설정하는 뷰잉 좌표계를 이용

뷰잉 변환

- **뷰잉 좌표계는 뷰 평면의 축 벡터와 법선 벡터를 이용하여 설정**
  - 원점: 뷰 평면 상의 기준점 (카메라 위치) → VRP: View Reference Point
  - Normal Vector: z축에 해당 (바라보는 방향) → VPN: View Plane Normal Vector
  - Up Vector: y축에 해당 (카메라 각도) → VUP: View Up Vector
    - x축은 자동으로 결정

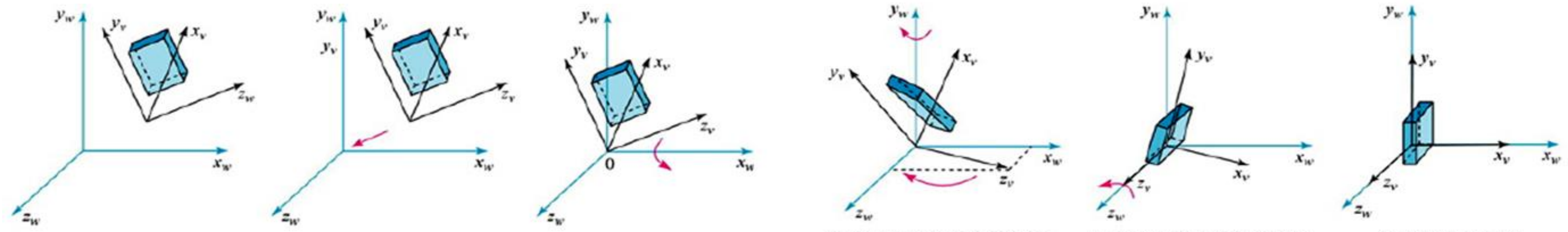


(a) 월드좌표계와 뷰평면 (b) 뷰평면에 나타난 객체

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

좌표계 변환

- **월드 좌표계에서 뷰잉 좌표계로 변환**
  a. 뷰잉좌표계와 월드좌표계가 주어짐
  b. 뷰잉좌표계 원점을 월드좌표계 원점과 일치하도록 이동
  c. 월드좌표계의 x축을 중심으로 뷰잉좌표계의 z축을 회전
     → 뷰잉좌표계의 z축이 월드좌표계의 zx 평면에 위치
  d. 월드좌표계의 y축을 중심으로 뷰잉좌표계를 회전
     → 두 좌표계의 z축이 일치
  e. 월드좌표계의 z축을 중심으로 뷰잉좌표계를 회전
  f. 뷰잉좌표계와 월드 좌표계가 일치



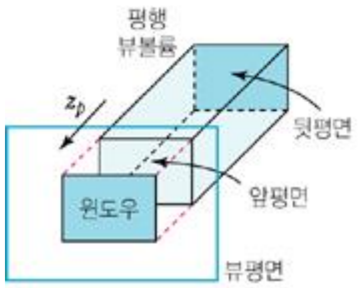
(a) 뷰잉좌표계와 월드좌표계의 위치
(b) 뷰잉좌표계를 이동
(c) 월드좌표계의 x축을 중심으로 뷰잉좌표계를 회전
(d) 월드좌표계의 y축을 중심으로 뷰잉좌표계를 회전
(e) 월드좌표계의 z축을 중심으로 뷰잉좌표계를 회전
(f) 일치된 두 좌표계

투영을 위한 변환

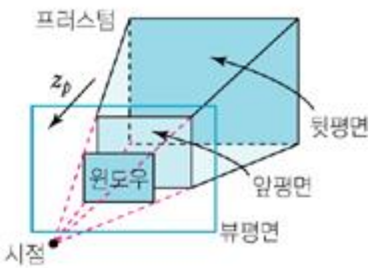
- **뷰평면의 윈도우 내에 투영되는 공간상의 일정영역**
  - 투영 변환에서 뷰평면의 윈도우에 투영되는 객체들은 3차원 공간에서 일정한 영역 내에 존재: 뷰볼륨
    - 평행 투영의 경우: 평행 뷰볼륨
    - 원근 투영의 경우: 프러스텀(Frustum) 뷰볼륨

  - 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



(a) 평행투영



(b) 원근투영

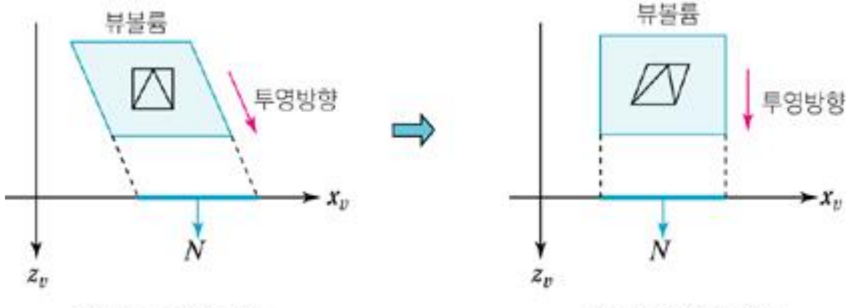
투영을 위한 변환

- **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,

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

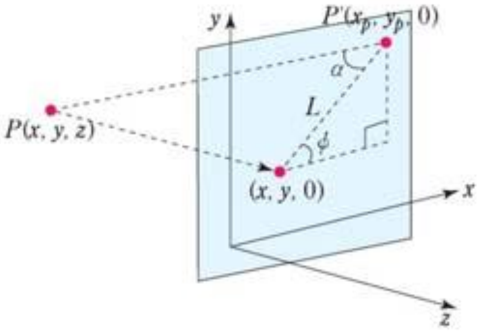
투영을 위한 변환

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

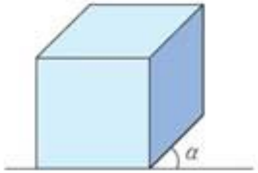


(a) 경사투영의 경우 (b) 직각투영의 경우

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



(a) 공간상의 한 점이 경사투영된 경우

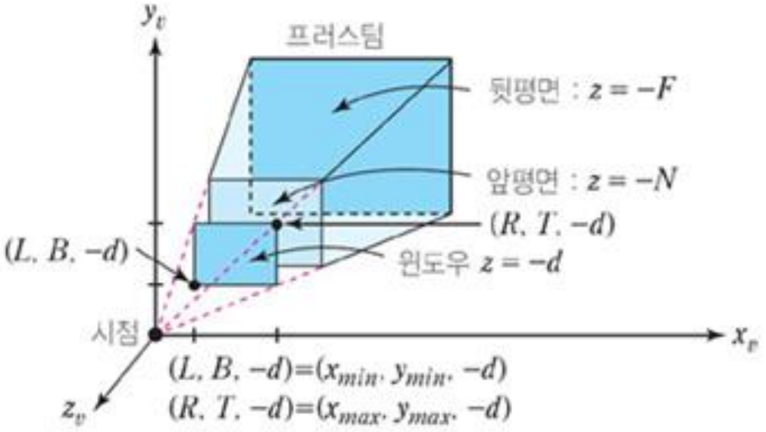


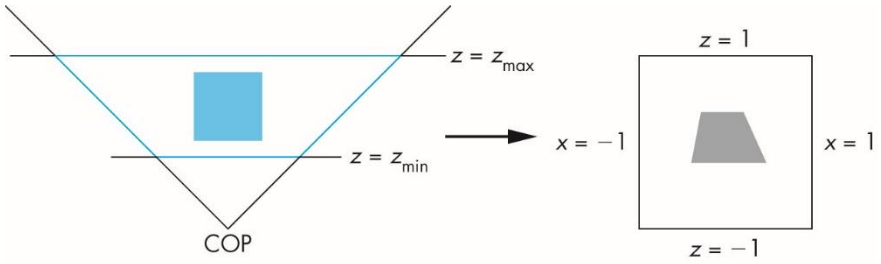
(b) 경사투영의 경우

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

투영을 위한 변환

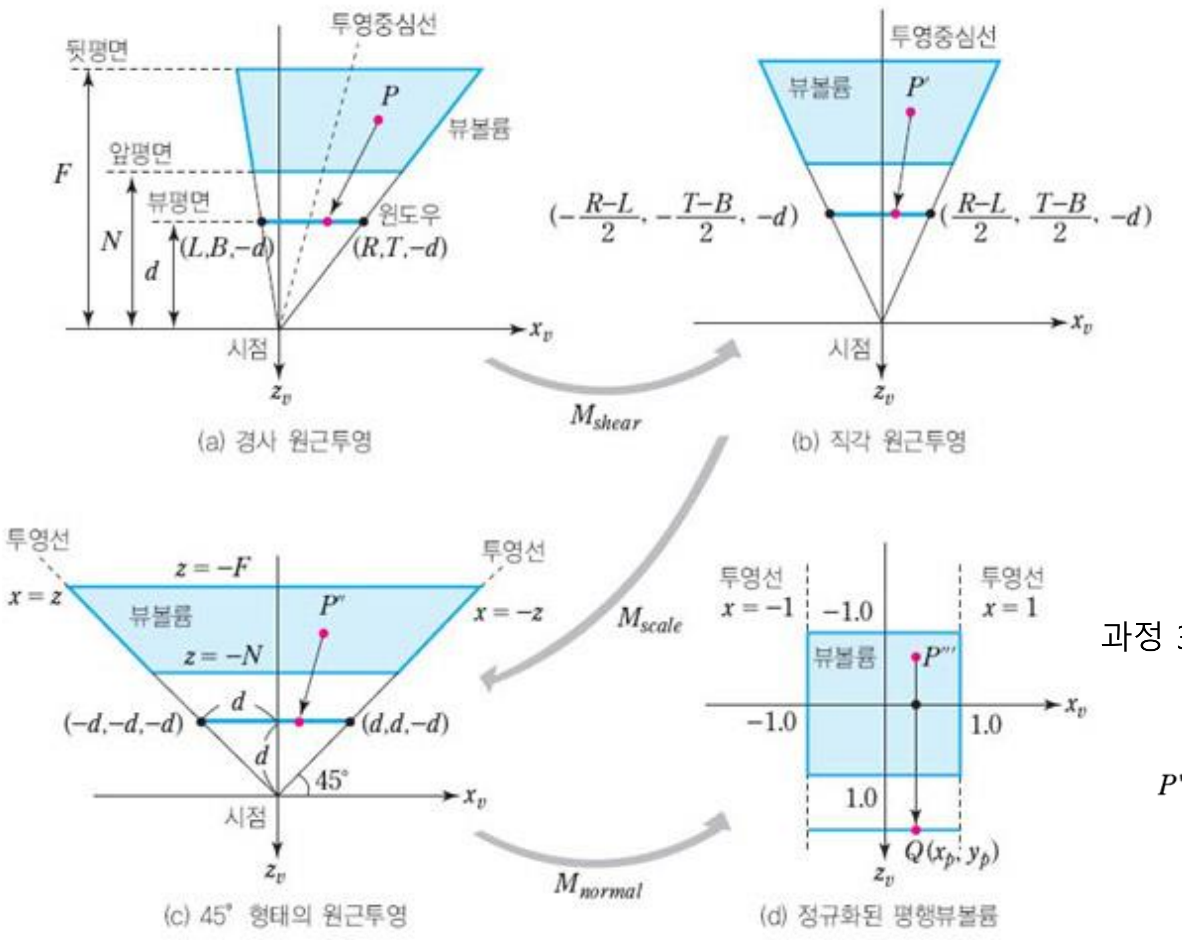
- **원근 투영의 변환 행렬**
  - 프러스텀을 직육면체 형태로 변환하여 직각투영 이용
    - 시점: 뷰잉 좌표계의 원점
    - 윈도우: 법선벡터는 z축 방향
    - 뷰평면 기준: left, right, top, bottom
      - d: 뷰 평면이 놓여진 z 값
    - 프러스텀 뒷 평면과 앞 평면: -F, -N





투영을 위한 변환

– 밀림변환과 신축변환을 수행



과정 1: 밀림변환 적용 (P → P′)

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

$$\left(-\frac{R-L}{2}, -\frac{T-B}{2}, -d\right) \quad \left(\frac{R-L}{2}, \frac{T-B}{2}, -d\right)$$

과정 2: 신축변환 적용 (P′ → P″)

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

과정 3: 정육면체 형태로 정규화 적용 (P″ → P‴)

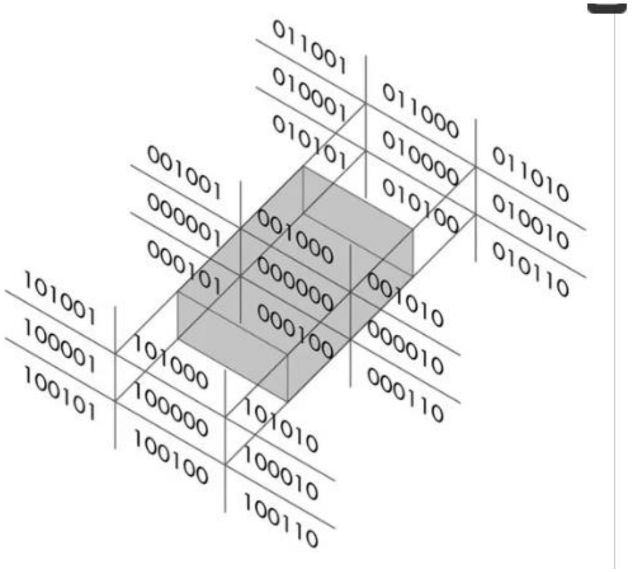
$$P''' = \begin{pmatrix} x''' \\ y''' \\ z''' \\ h \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{F+N}{F-N} & -\frac{2FN}{F-N} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x'' \\ y'' \\ z'' \\ 1 \end{pmatrix} = M_{normal} \cdot P''$$

3D 클리핑 알고리즘

- 3D Cohen-Sutherland 라인 클리핑
  – 6비트 코드를 사용하여 선의 끝점을 분류한다

비트1 앞 (front)	비트 2 뒤 (behind)	비트 3 위 (above)	비트 4 아래 (below)	비트 5 오른쪽 (right)	비트 6 왼쪽 (left)

1) 양 끝 점이 모두 000000 이면 → 그린다
2) 양 끝점 중 한 개는 000000이고 다른 한 개는 0이 아니면 → 일부를 그린다.
3) 양 끝점이 모두 0이 아니고 AND 연산이 0이 아니면 → 안 그린다.
4) 양 끝점이 모두 0이 아니고 AND 연산이 0이면 → 클리핑



3D 클리핑 알고리즘

- **3D Liang-Barsky 선 클리핑 알고리즘**
  - 선의 시작점 (x0, y0, z0), 끝점: (x1, y1, z1)
  - 선의 매개변수 방정식

    x = x0 + (x1 - x0)u

    y = y0 + (y1 - y0)u

    z = z0 + (z1 - z0)u

    - 매개변수 t: 0 ≤ t ≤ 1

  - 클리핑 조건은
    - xmin ≤ x0 + (x1 - x0)u ≤ xmax
    - ymin ≤ y0 + (y1 - y0)u ≤ ymax
    - zmin ≤ z0 + (z1 - z0)u ≤ zmax

➔ pk · u ≤ qk (k = 1, 2, 3, 4, 5, 6)

$p1$= -(x1 - x0) , q1= (x0 - xmin) (왼쪽 경계)

$p2$= (x1 - x0) , q2= ($x$max - x0) (오른쪽 경계)

$p3$= -(y1 - y0) , q3= (y0 - $y$min) (하단 경계)

$p4$= (y1 - y0), q4= ($y$max - y0) (상단 경계)

$p5$= - (z1 - z0) , q5= (z0 - $z$min) (먼 쪽 경계)

$p6$= (z1 - z0) , q6= ($z$max - z0) (가까운 쪽 경계)

3D 클리핑 알고리즘

- pk = 0 → 선이 클리핑 영역에 평행
  - pk = 0, qk< 0 → 클리핑 영역 외부
  - pk = 0, qk > 0 → 클리핑 영역 내부
- pk < 0 → 선은 영역 외부에서 내부로 이동
- pk > 0 → 선은 내부에서 외부로 이동

pk < 0 ➔ u1 = max (0, $\frac{qk}{pk}$) (k = 1, 2, 3, 4, 5, 6)

pk > 0 ➔ u2 = min (1, $\frac{qk}{pk}$) (k = 1, 2, 3, 4, 5, 6)

1) u1 > u2 ➔ 선은 영역 외부에 있어 그리지 않는다.
2) u1 < u2 ➔ 일부가 영역 안에 있어서 u1과 u2를 사용하여 선의 새로운 끝점을 계산한다.

n_x0 = x0 + u1∗dx　　n_y0 = y0 + u1∗dy　　n_z0 = z0 + u1∗dz

n_x1 = x0 + u2∗dx　　n_y1 = y0 + u2∗dy　　n_z1 = z0 + u2∗dz

이번 주에는

- **투영**
  - 평행투영
  - 원근투영
- **뷰잉 변환**
  - 3차원 클리핑 알고리즘

- **다음에는**
  - 3차원 객체: 다각형
  - 스플라인