OpenGL 9

가상현실론

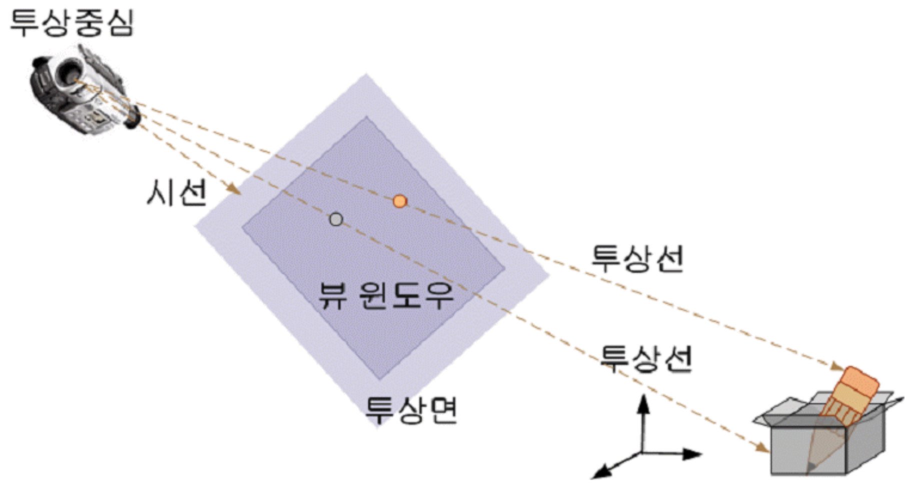
2021/03/24

목차

- 복습
  - 투상
  - GL의 투상변환
- 정규화 가시부피
- GL의 원근투상
- 정규화 가시부피에 의한 원근투상
- 원근투상의 정규화 전환

복습

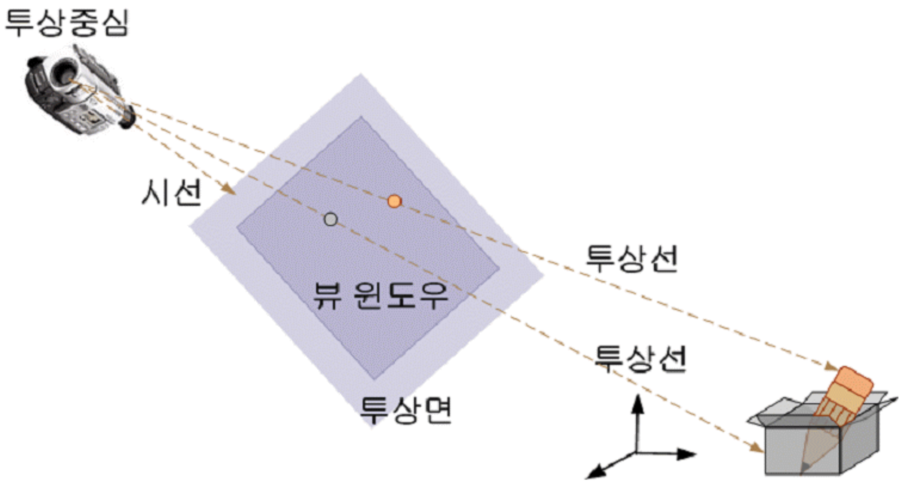
- 투상



- 시선 (Line of Sight): 카메라가 바라보는 방향. 카메라 설정 방법에 따라서 시선은 전역좌표계 원점을 향하기도 하고, 임의 위치의 초점을 향하기도 함
- 투상면 (Projection Plane, View Plane): 물체 영상이 맺히는 화면 (영화관 스크린과 같은 역할). 일반적으로 시선에 수직으로 놓인다.
- 투상중심에서 물체 정점을 향한 투상선이 투상면과 만나는 곳에 해당 정점이 투상(Projection)됨.
- 투상은 평행 투상과 원근 투상으로 구분

복습

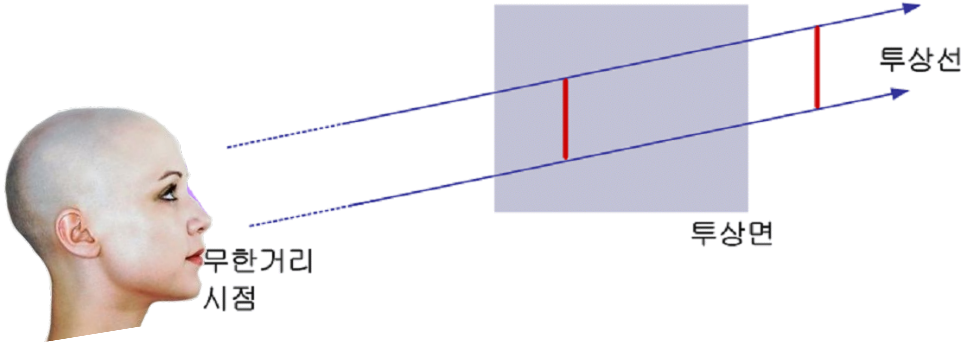
- 투상



- 시선 (Line of Sight): 카메라가 바라보는 방향. 카메라 설정 방법에 따라서 시선은 전역좌표계 원점을 향하기도 하고, 임의 위치의 초점을 향하기도 함
- 투상면 (Projection Plane, View Plane): 물체 영상이 맺히는 화면 (영화관 스크린과 같은 역할). 일반적으로 시선에 수직으로 놓인다.
- 투상중심에서 물체 정점을 향한 투상선이 투상면과 만나는 곳에 해당 정점이 투상(Projection)됨.
- 투상은 평행 투상과 원근 투상으로 구분

복습

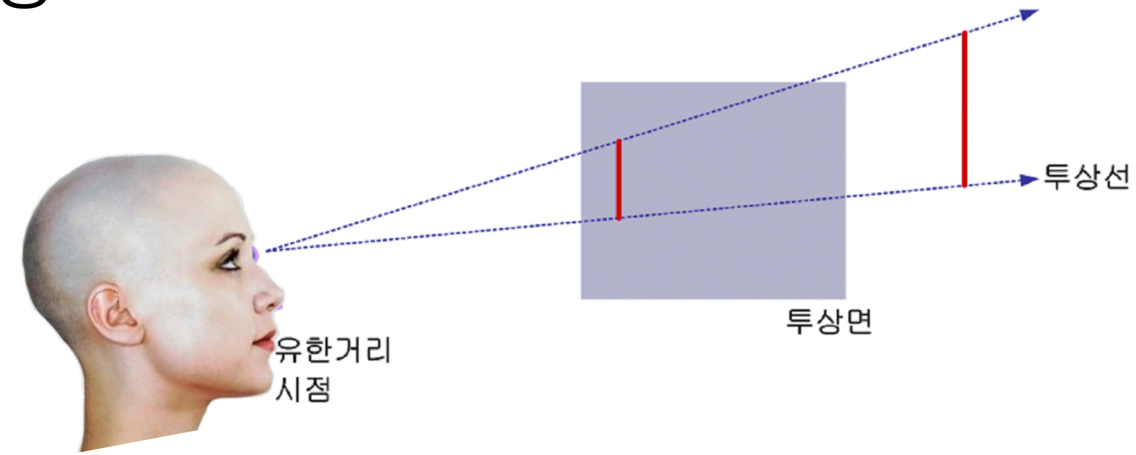
- 투상-평행투상



- 평행 투상(Parallel Projection): 시점이 물체로부터 무한대의 거리에 있다고 간주하여 투상선을 나란히 가져가는 투상법.
- 태양이 지구로부터 매우 먼 거리에 위치해있기 때문에 태양에서 출발한 빛은 지구에 도착할 때 평행광선이라는 것에 비유할 수 있음.
- 평행 투상에서 원래 물체의 평행선은 투상 후에도 평행.
- 평행투상은 정사 투상, 축측 투상, 경사 투상으로 구분됨

복습

- 투상-원근투상

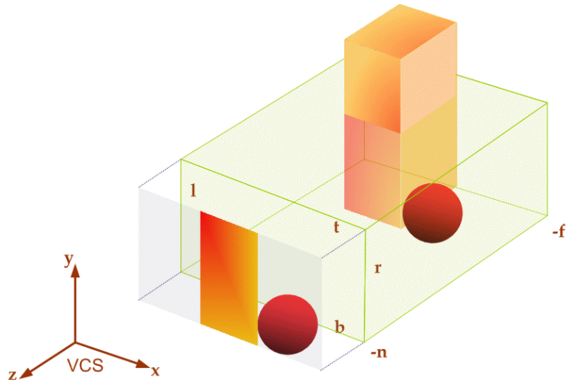


- 원근투상 (Perspective Projection): 시점이 물체로부터 유한한 거리에 있다고 간주하여 모든 투상선이 시점에서 출발하여 방사선 모양으로 퍼져가는 방법.
- 카메라나 사람의 눈이 실제로 물체를 포착하는 방법

복습

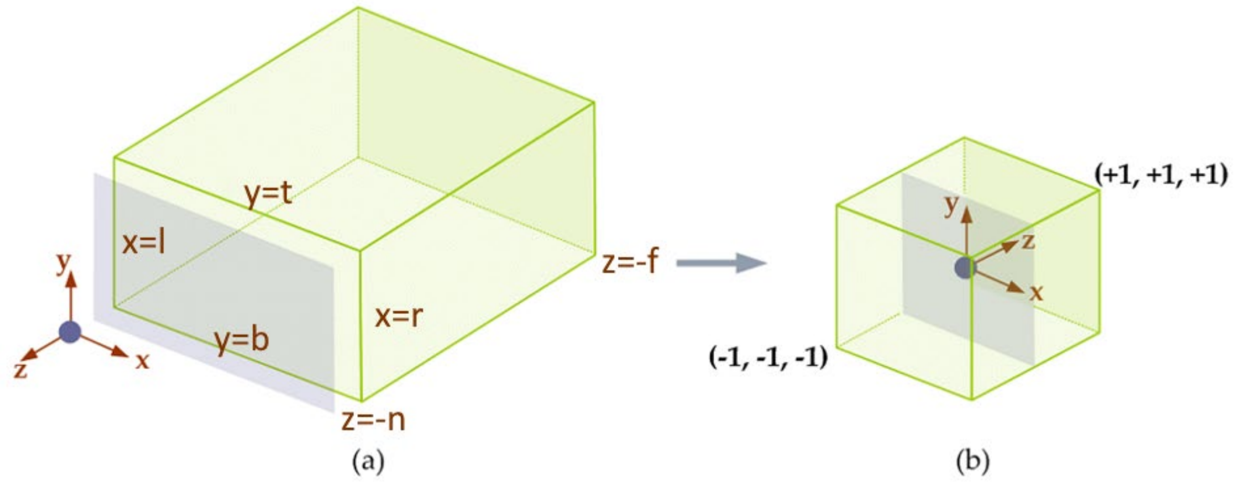
- GL의 투상변환
  - 투상행렬(Projection Matrix): 투상 변환을 행렬로 표시한 것 (모델 변환과 시점 변환을 행렬로 표시한 것은 모델 뷰 행렬)
  - void glMatrixMode(GL_PROJECTION)
    - 행렬 모드를 투상 행렬로 설정하는 함수
  - 행렬 모드를 설정하지 않을 경우의 기본값은 모델 뷰 행렬 (GL_MODELVIEW). 투상을 가하기 위해서는 반드시 이 함수를 호출해야 함.
  - 호출 이후 다시 행렬 모드를 바꾸기 전까지 제시되는 모든 변환 함수는 투상 행렬에 반영.
  - Projection matrix은 model-view matrix와 마찬가지로 matrix stack으로 구성.
  - 투상 행렬 스택(Projection matrix stack)은 최소한 2개 이상. 스택의 맨 위에 있는 것이 현 변환 행렬(CTM_P: Current Transformation matrix). Model-view matrix의 CTM_M과는 완전히 다른 것임에 유의
  - void glGetFloatv(GLenum pname, GLfloat *params);
    - 현 투상 행렬 CTM_P를 검색할 때 사용.

정규화 가시 부피



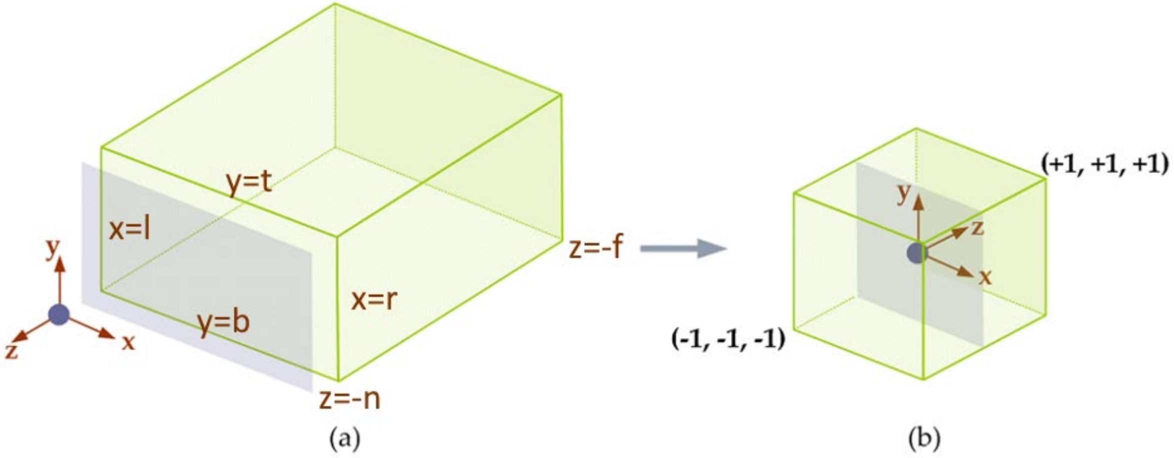
- 가시 부피 안에 있는 물체만 투상
- void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);
  - 평행 투상의 가시 부피를 설정. 정사 투상보다는 일반적인 평행 투상을 위한 함수 투상면이 물체의 주 평면과 반드시 나란할 필요는 없기 때문.
  - 파라미터는 부피의 좌, 우, 하, 상, 전, 후를 뜻함.
  - near와 far값은 반드시 <u>**양수**</u>로 나타내야 함. 시점으로부터 절단면까지의 거리이기 때문. 다만 실제 z좌표는 -1를 곱해야 함 (예) 전방 절단면 우상단 좌표 (right, top, -near))

정규화 가시 부피



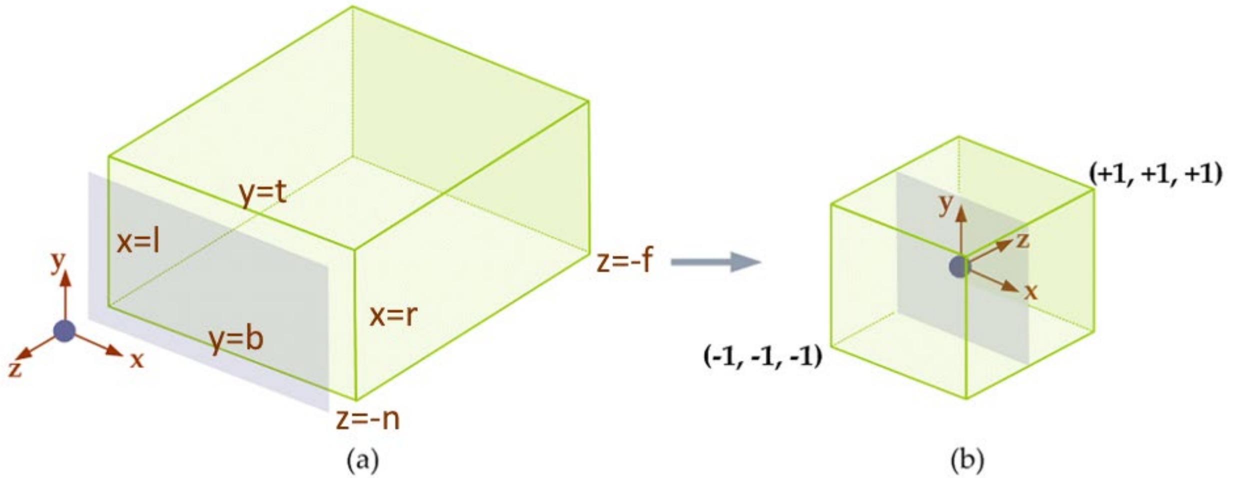
- 정규화 가시부피 (CVV: Canonical View Volume)
  - 가로, 세로, 높이가 2인 정육면체로 투상
  - 정규화 변환 (Normalization Transformation)

정규화 가시 부피



- 제한된 크기의 가시 부피를 제한된 크기의 뷰 윈도우로 투상하기 위해서 GL은 정규화 변환(Normalization Transformation)을 통해서 정규화.
- GL이 가시 부피 그대로를 사용하지 않는 이유는
  - 평행 투상, 원근 투상 모두 동일한 모습의 정규화 가시 부피를 사용함으로써 파이프라인 처리 구조가 동일
  - 가시 부피 밖의 물체를 절단하는 데 있어서 정규화 가시 부피인 정육면체를 기준으로 하는 것이 원래의 가시 부피를 그대로 절단하는 것보다 단순
  - 시점 좌표계 원점을 중심으로 가로, 세로 길이를 1로 정규화함으로써 화면 좌표계로 변환하기가 수월

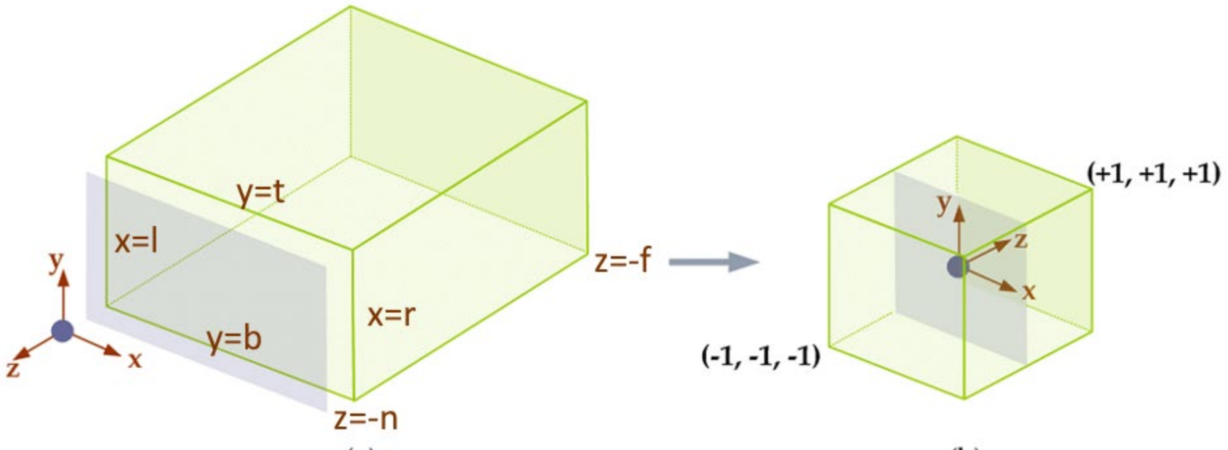
정규화 가시 부피



- 좌측의 가시 부피를 우측의 시점 좌표계 원점으로 이동하기 위한 이동 변환 행렬(translational transformation matrix)

$$T = \begin{pmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

정규화 가시 부피

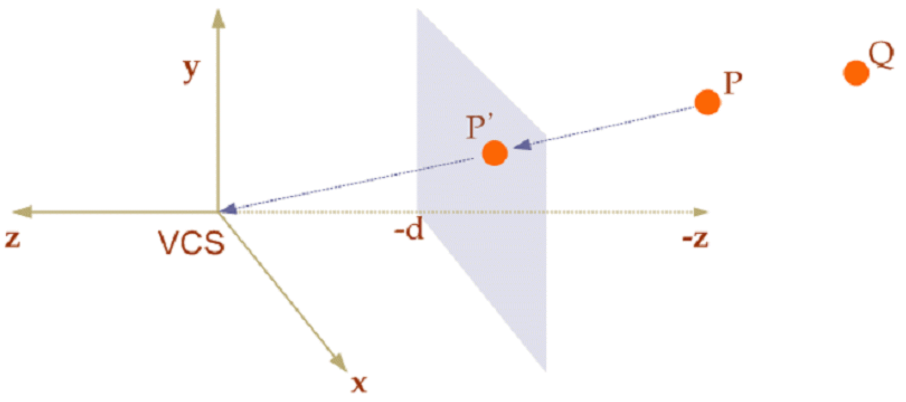


(a) (b)

- 이동된 가시부피에 크기조절을 가하여 길이 2인 정육면체를 만들기 위한 크기 조절 변환 S
- 왼손 좌표계로 변경함에 따른 z축 방향 반전을 위한 반사 변환 R

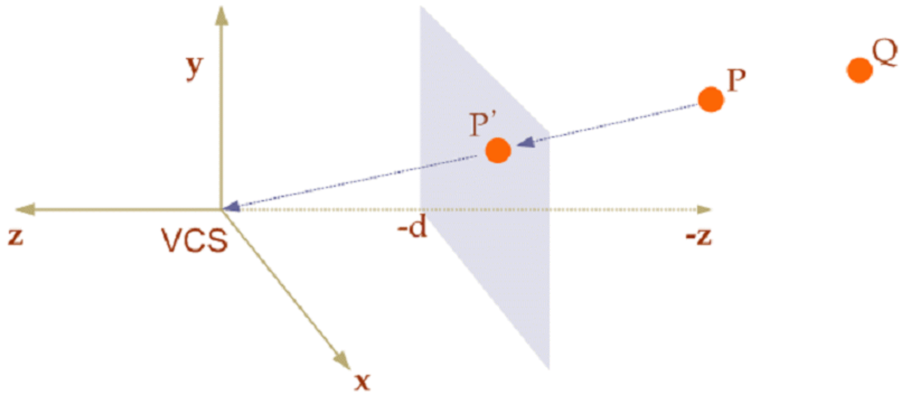
$$S = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad N = R \cdot S \cdot T$$

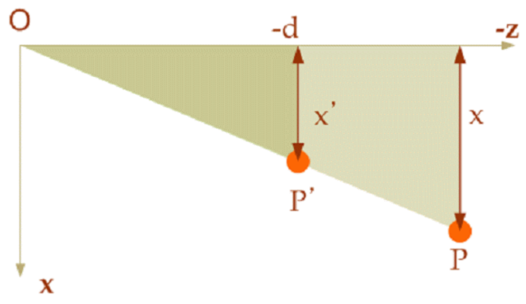
GL의 원근투상



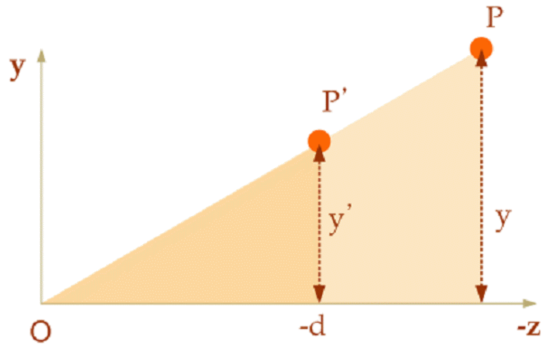
- 기본투상
  - P를 P’으로 투상
  - 시점좌표계 기준으로 P는 (x, y, -z)에 위치.
  - P’은 z=-d인 투상면에 위치하므로 P‘=(x’, y’, -d)

GL의 원근투상





(a)



(b)

- x' : (-d) = x : (-z), y' : (-d) = y : (-z)

$$\mathrm{P'} = \begin{pmatrix} x' \\ y' \\ -d \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ -d \\ 1 \end{pmatrix}$$

$$\mathrm{P'} = \begin{pmatrix} x' \\ y' \\ -d \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ -z \\ z/d \end{pmatrix}$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

GL의 원근투상

- 원근분할(Perspective Division, Homogenization)
  - 동차좌표의 마지막 요소로 이전 요소를 나누는 작업
  - 절단이 동차좌표에서 이루어지기 때문에 절단 이후로 미루어 짐.

$$\begin{pmatrix} x \\ y \\ -z \\ z/d \end{pmatrix} \longrightarrow \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ -d \\ 1 \end{pmatrix}$$

$$\mathrm{P'} = \begin{pmatrix} x' \\ y' \\ -d \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ -d \\ 1 \end{pmatrix}$$

  - z가 커질 수록 (멀어질 수록) z 성분은 없어지고 x, y 성분만 남게 됨을 의미.
  - 또한 x, y축 성분 또한 멀어질 수록 작아짐.
  - 즉, 멀리 있는 물체일수록 작어보임.

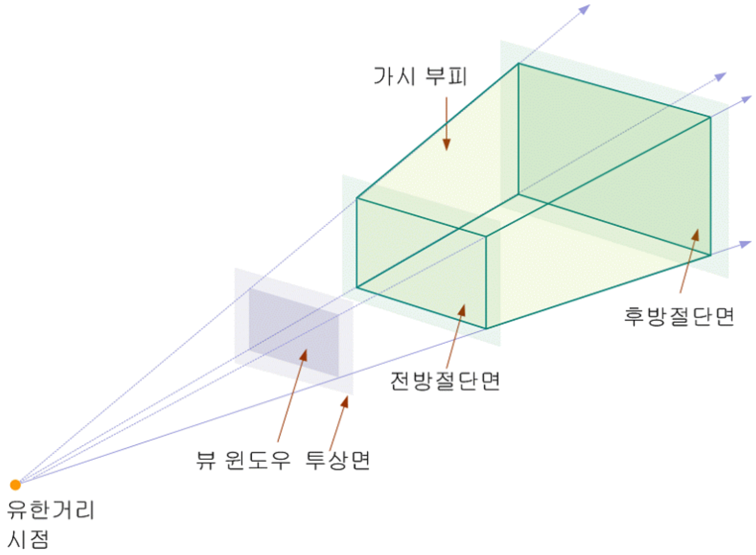
GL의 원근투상

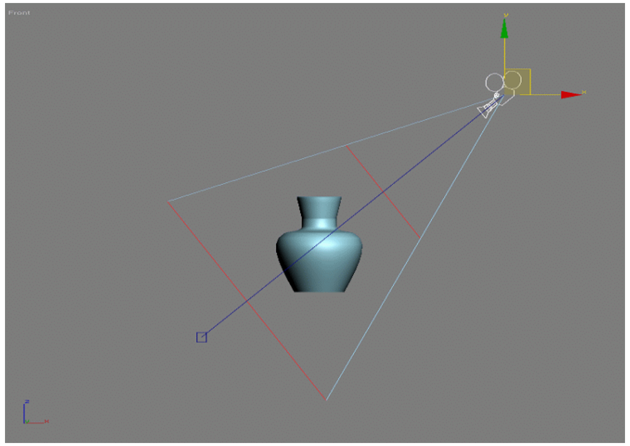
$$\begin{pmatrix} x \\ y \\ -z \\ z/d \end{pmatrix} \longrightarrow \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ -d \\ 1 \end{pmatrix}$$

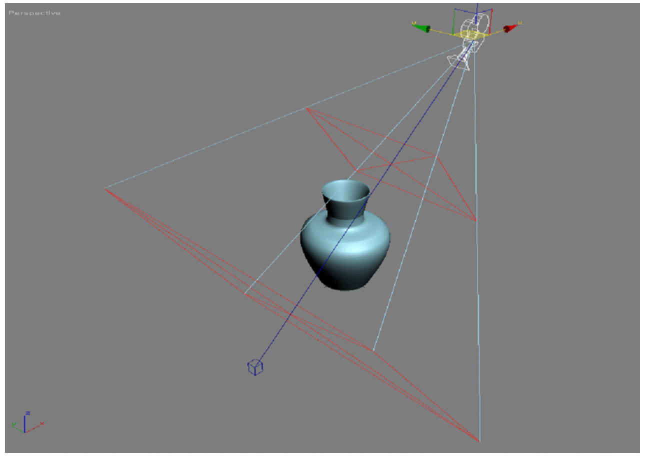
- 원근변환
  - 어파인 변환이 아님: 마지막 행이 (0, 0, 0, 1)이 아님
  - 3차원 좌표관점: x' = x/(z/d): 비선형 변환
  - 4차원 동차좌표 관점: 선형변환

$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \mathrm{Mperspective} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

정규화 가시부피에 의한 원근투상

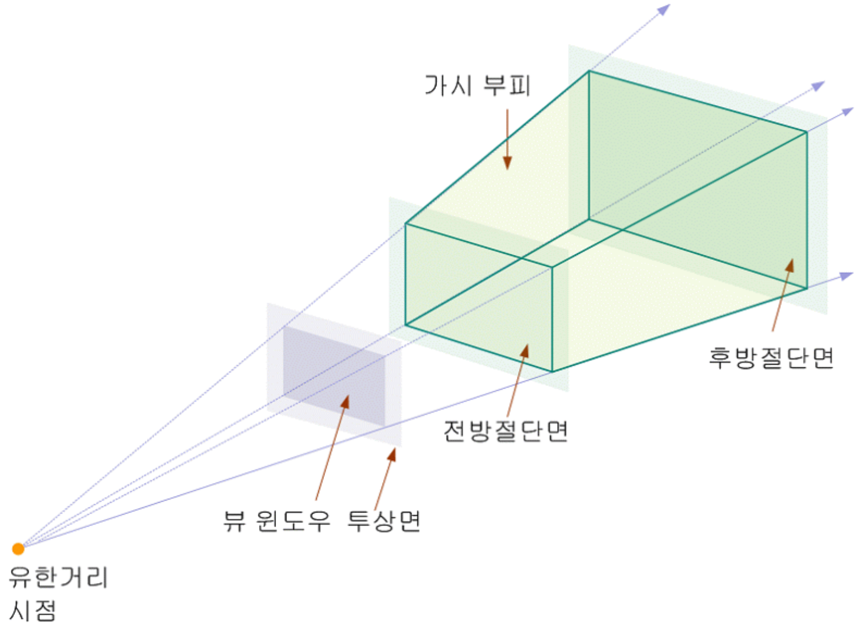


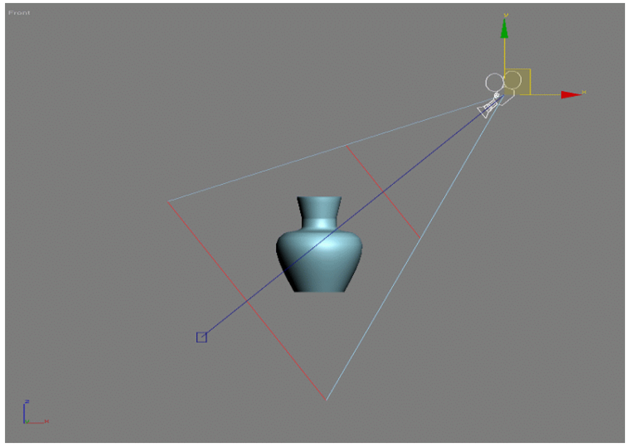




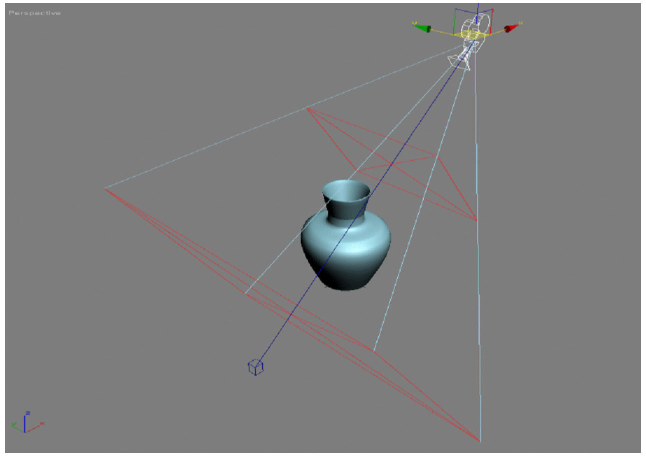
- 평행 투상과 마찬가지로 GL의 원근 투상에서도 항상 투상 범위를 제한해야 함.
- 원근 투상에서 투상면과 전후방 절단면은 시점으로부터 초첨을 향한 선, 즉 시선에 수직으로 놓임
- 절단 사각뿔(Frustum) = 절두체 형태의 가시 부피

정규화 가시부피에 의한 원근투상



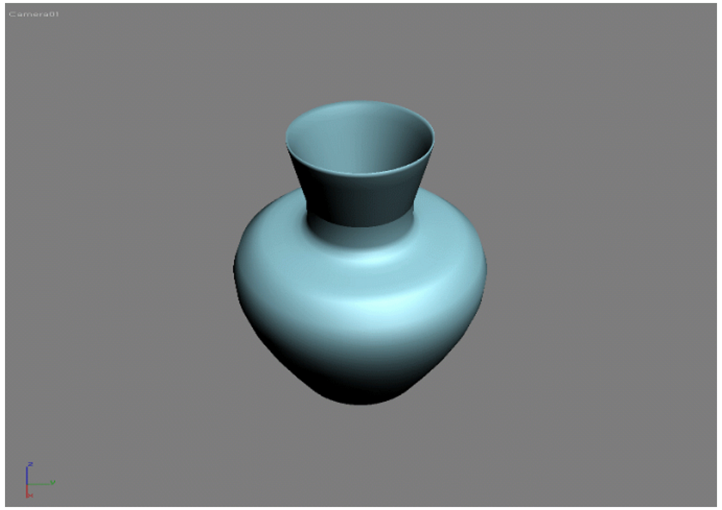


가시부피 1

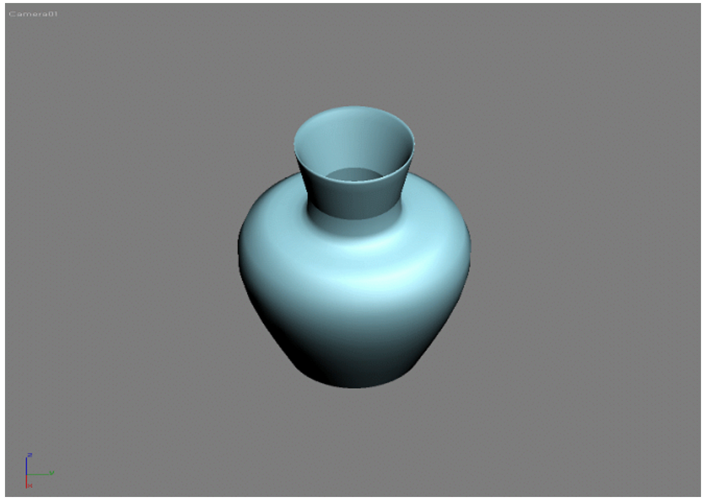


가시부피 2

정규화 가시부피에 의한 원근투상



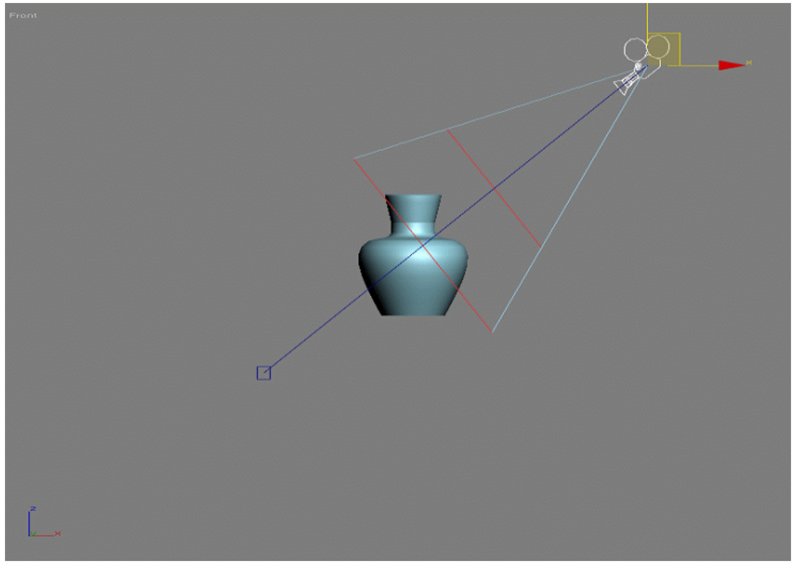
원근투상

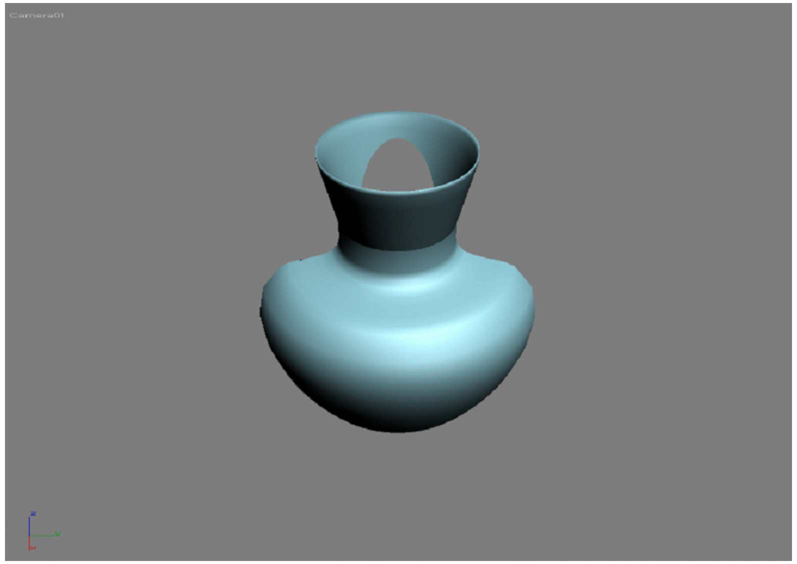


평행투상

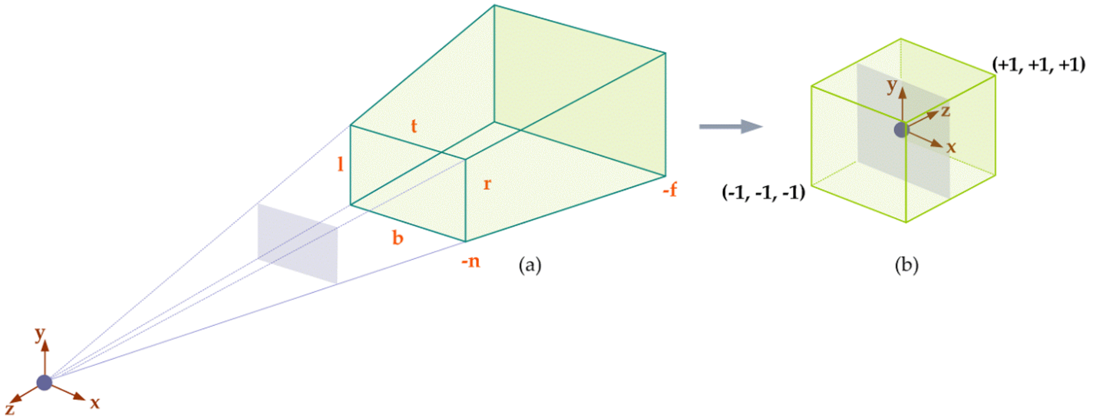
정규화 가시부피에 의한 원근투상

- 가시부피 설정에 의한 절단





원근투상의 정규화 가시부피

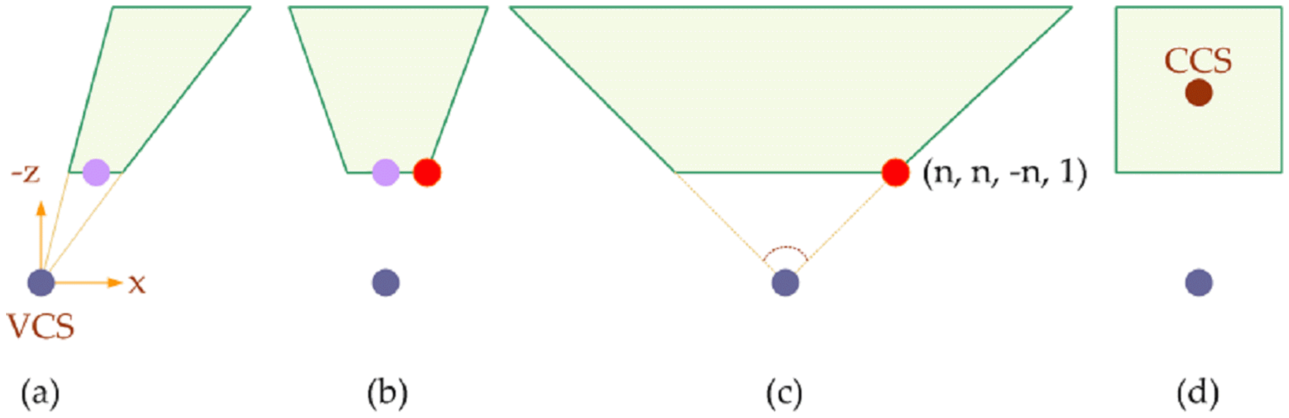


- GL의 원근 투상의 가시 부피는 평행 투상과 동일한 모습의 정규화 가시 부피로 변환
- 원근 투상 가시 부피 중 전후방 절단면은 각각 절단좌표계에서 z=-1인 평면과 z=1인 평면으로 사상.
- 투상면은 절단 좌표계의 z=0인 평면에 놓이게 됨.
- 정규화 가시부피로 변환되면 GL은 평행투상과 원근 투상을 동일한 방식으로 취급.

GL의 원근투상

- void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);
  - GL의 원근 투상에서 가시 부피의 좌우상하전후를 정의함.
  - gluPerspective와 이 함수가 다른 점은 가시 부피의 위치. gluPerspective()에서는 시선이 정확히 가시 부피의 중심을 통과하며, 가시 부피가 시선을 기준으로 대칭적으로 놓임. glFrustum() 함수는 이러한 제약 조건이 없음.

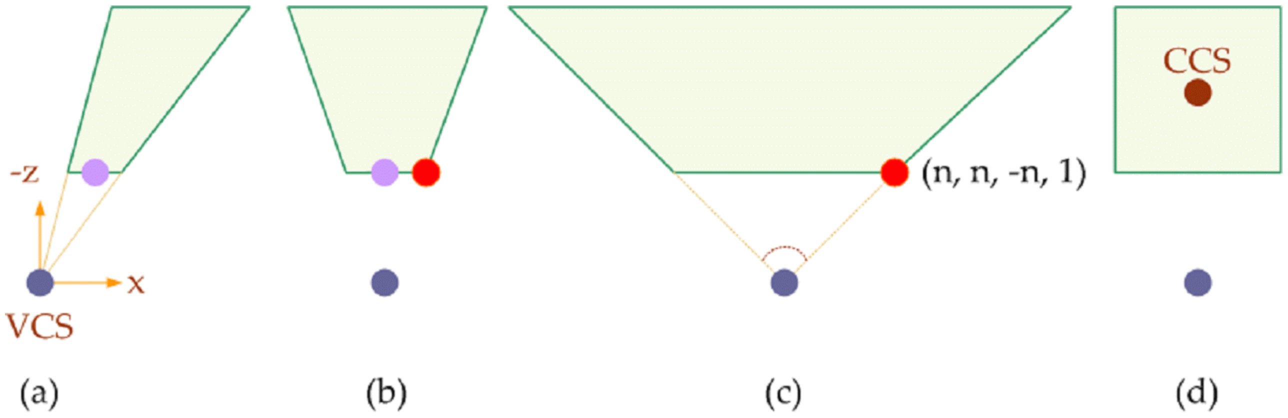
glFulstum()의 가시부피



<가시 부피를 위에서 -y축 방향으로 내려본 것>

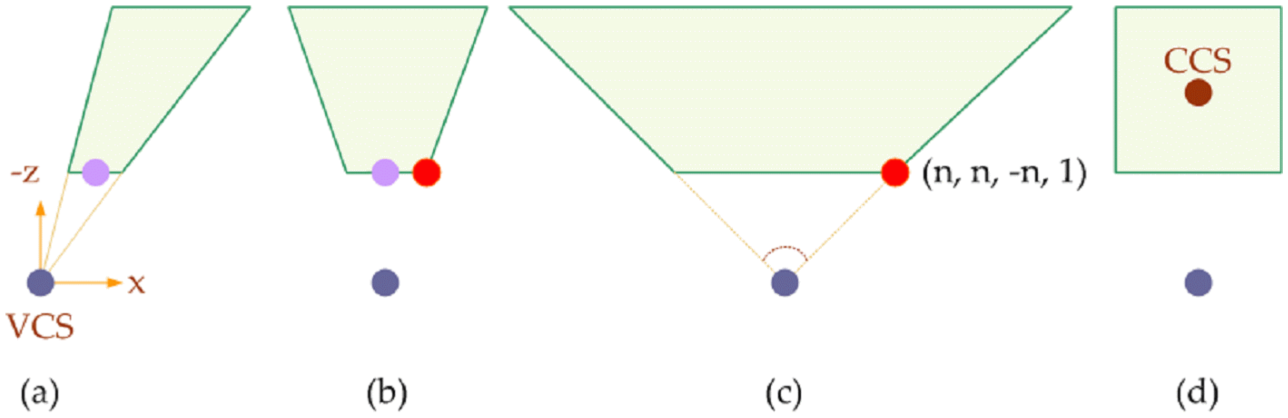
- (a)를 보면 가시 부피가 시선방향인 –z축에서 벗어나 있음. 경사투상(Oblique Projection)과 유사한 모습을 얻기 위함 (원근투상이기 때문에 경사투상과는 다른 결과)

원근 투상의 정규화 변환(Perspective Normalization)



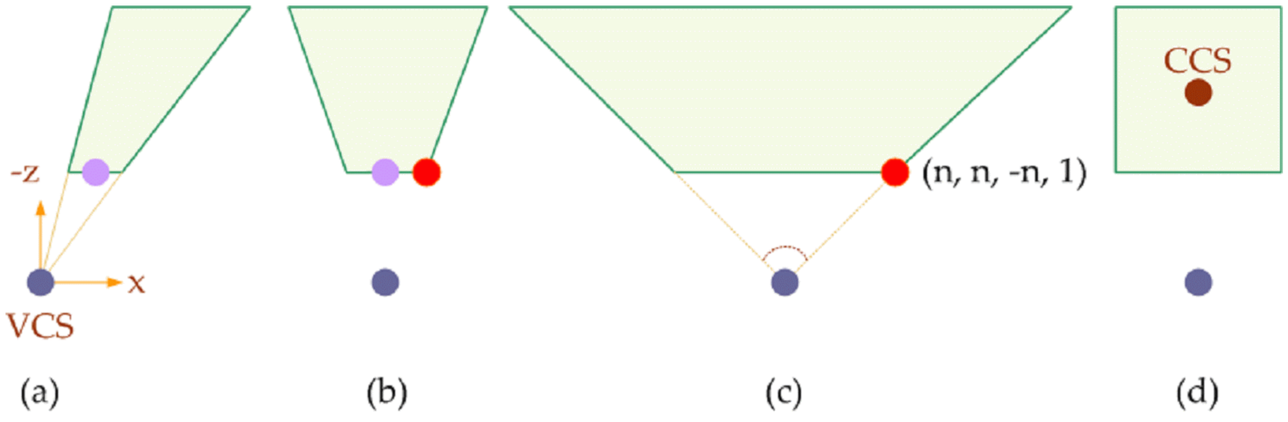
- (a)~(d): 원근 투상을 정규화 변환하는 과정
- 우선 (a)의 부피를 (b)와 같이 z축을 기준으로 대칭적인 모습으로 만들기 위해서 (a)의 가시부피를 x-y 평면에서 왼쪽으로 밀어내는 전단(shearing) 변환을 가함
- Sh= $\begin{bmatrix} 1 & 0 & \frac{r+l}{2n} & 0 \\ 0 & 1 & \frac{t+b}{2n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$
- 위의 전단 변환은 (a)의 전방절단면 중심인 ((r+l)/2, (t+b)/2, -n, 1)을 (b)의 전방 절단면 중심인 (0, 0, -n, 1)로 밀어내기 위함임

원근 투상의 정규화 변환(Perspective Normalization)



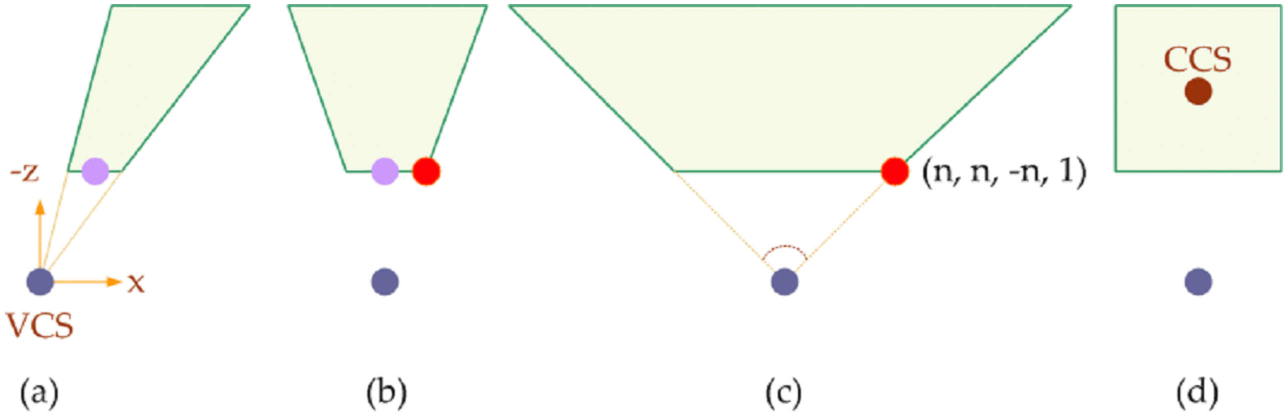
- (b)에서 (c)로의 변환은 시점으로부터 전방 절단면까지의 거리 n을 절단면 크기에 반영하기 위한 작업으로, 전방 절단면 가로 세로의 길이가 2n이 되도록 고정.
- (c)의 가시 부피 우측면에 대해서는 x=-z, 좌측면에 대해서는 x=z 성립.
- (b)의 전방 절단면 우상단 ((r-l)/2, (t-b)/2, -n, 1)을 (c)의 (n, n, -n, 1)로 바꾸는 크기 조절 변환행렬은,
- S= $\begin{bmatrix} \frac{2n}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2n}{t-b} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$

원근 투상의 정규화 변환(Perspective Normalization)



- (c)를 (d)로 바꾸는 변환행렬은 다음과 같음
- T= $\begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & \frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{bmatrix}$

원근 투상의 정규화 변환(Perspective Normalization)

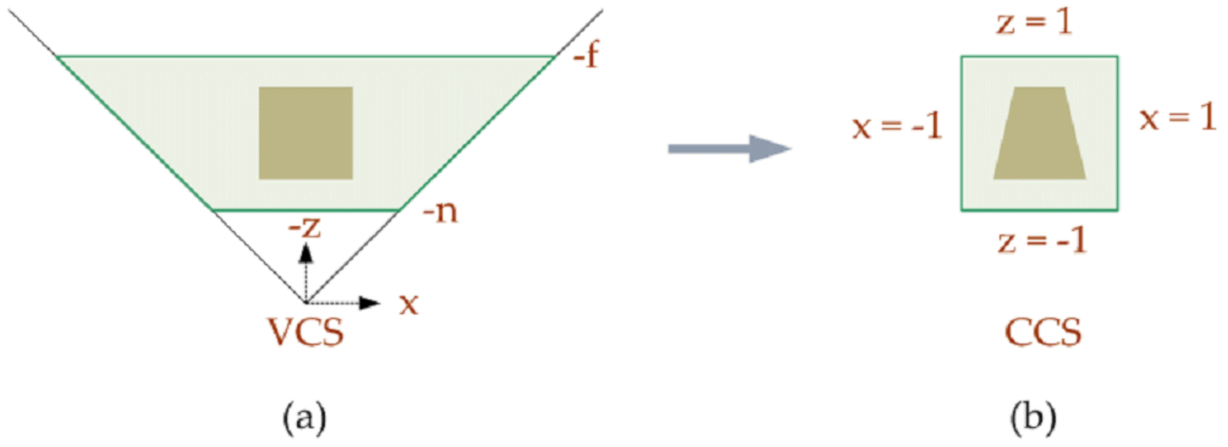


$$N = T \cdot S \cdot Sh$$

$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} \frac{2n}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2n}{t-b} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \frac{r+l}{2n} & 0 \\ 0 & 1 & \frac{t+b}{2n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

$$= \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

원근 투상의 정규화 가시부피



- 정규화 변환에 의해 가시 부피가 줄어들면서 부피 내의 물체의 모습이 변환됨
- (a)의 정육면체는 정규화 변환의 결과 (b)처럼 변형됨.
- 전방 절단면에 비해 후방 절단면이 크게 줄어든 것처럼 육면체 뒷면이 전면보다 크게 줄어듬.
- -> 원근 투상에서 멀리 있는 것이 작게 보이는 것이 기하변환을 통해서 구현됨