그래픽스 강의노트 05 - OpenGL 카메라

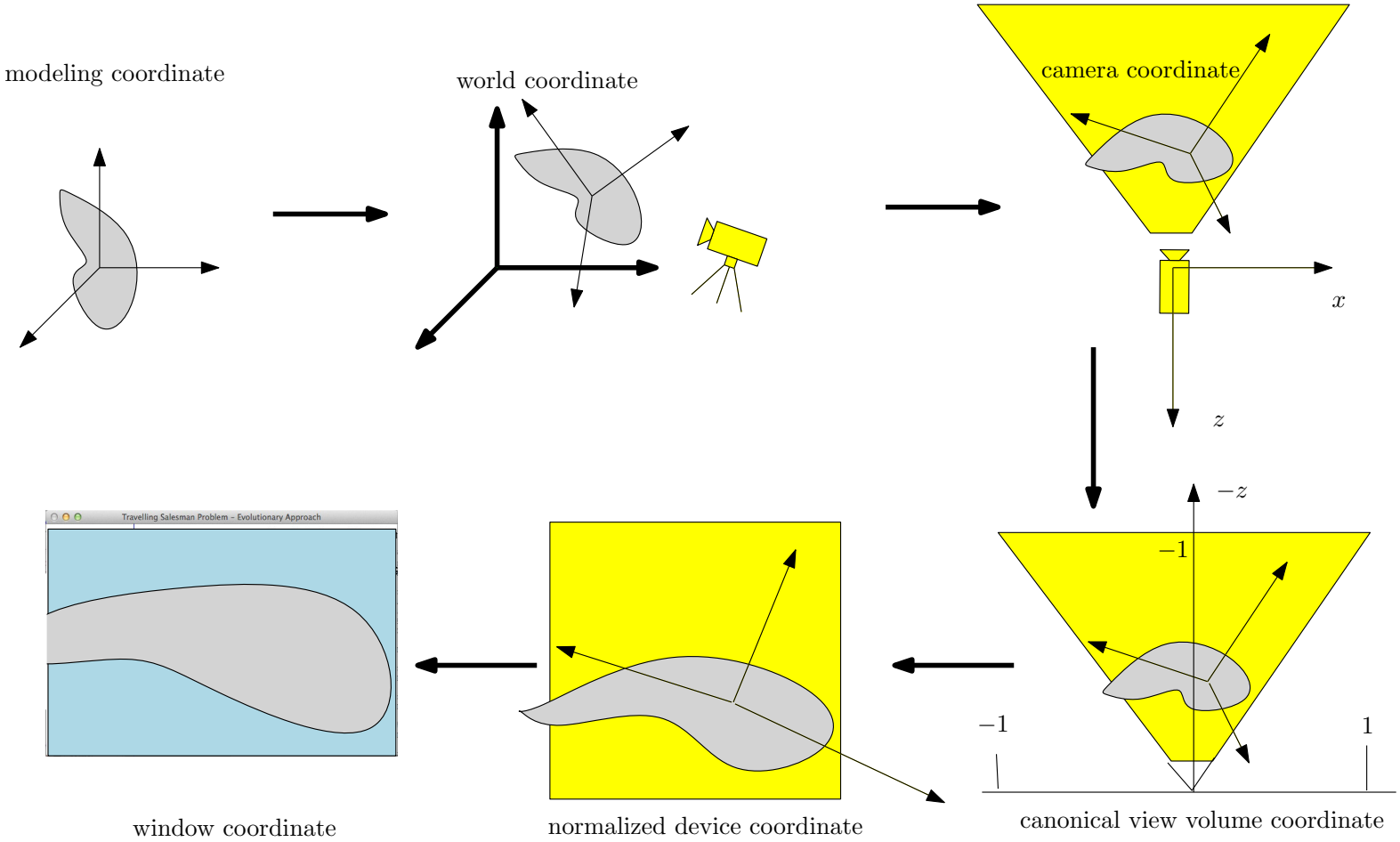
강영민

동명대학교

2015년 2학기

카메라 좌표 1/2

- 3차원 그래픽스에서 모든 정점은 카메라를 기준으로 좌표가 재배치
- OpenGL에서 이 카메라 좌표계의 원점은 카메라의 위치가 되고, 카메라가 바라보는 방향이 z축의 음의 방향

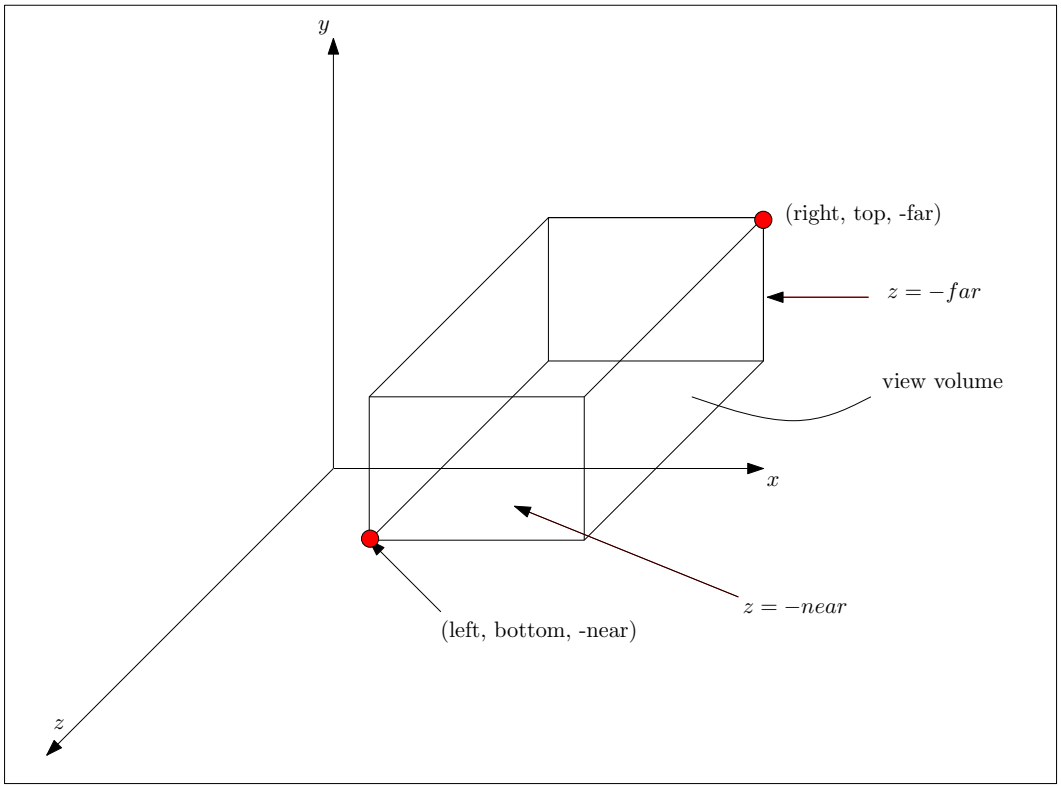


카메라 좌표 2/2

- 모델링 좌표계를 기준으로 객체를 구성하는 각 정점의 좌표 결정
- 객체에 적용되는 각종 변환들에 의해 전역 좌표(world coodinate)가 결정
- 렌더링 파이프라인은 이를 화면에 출력하기 위해 카메라 좌표계로 변경
  - 카메라의 위치가 원점이 되고, 카메라는 z축 음의 방향을 향함
- 관측볼륨의 변환
  - 관측 볼륨 밖은 처리에서 제외
  - 관측 볼륨은 클리핑 등이 효율적으로 이루어질 수 있는 표준 관측 볼륨(canonical viewing volume)으로 변환
  - 표준 관측 볼륨은 정규 장치 좌표계로 변경: x, y, z의 범위가 모두 $[-1, 1]$인 정육면체 공간
- 투영: 정규 장치 좌표에서 z 값을 버리고 (x, y)만 취함

직교 투영 카메라 1/4

- 디폴트 카메라의 경우 중심이 원점이고, 각 변의 길이가 2인 상자 모양인데, 이 위치와 길이를 변경할 수 있다. 이를 지원하는 함수는 glOrtho
- glOrtho(float left, float right, float bottom, float top, float near, float far);



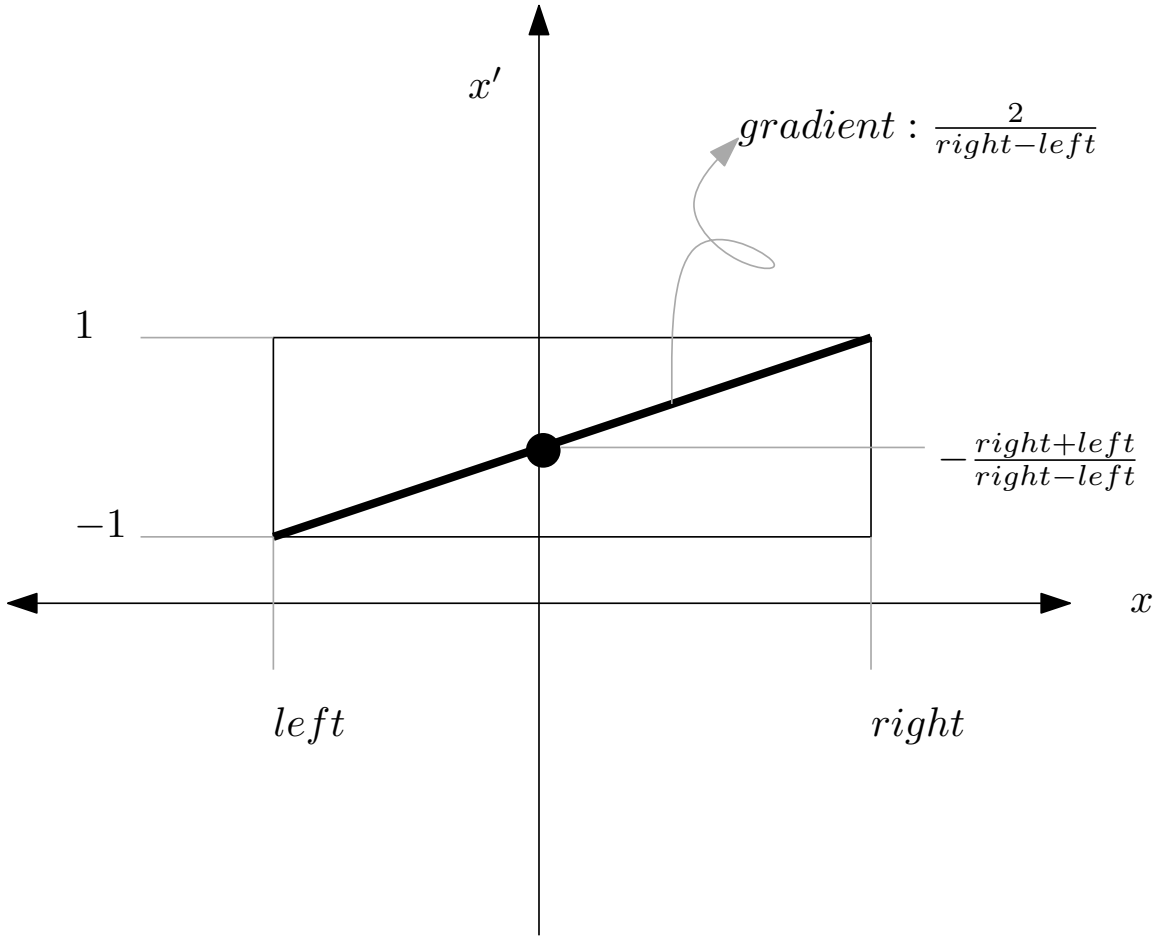
직교 투영 카메라 2/4

- OpenGL은 관측공간(view volume)을 최초의 디폴트 카메라 공간으로 옮기는 변환을 수행
- 관측공간 내부의 모든 객체가 디폴트 관측 공간인 (-1,1)×(-1,1)×(-1,1)로 옮겨지도록 하는 것
  - '정규 관측 공간(canonical view volume)'
- x 축으로는 범위 $[left, right]$를 $[-1, 1]$로 변경하는 것

$$x' = \left(\frac{2}{right - left}\right) x - \frac{right + left}{right - left}$$

직교 투영 카메라 3/4

$$x' = \left(\frac{2}{right - left} \right) x - \frac{right + left}{right - left}$$



직교 투영 카메라 4/4

비슷한 방법으로 y 축 좌표의 변경도 가능하다.

$$y' = \left(\frac{2}{top - bottom}\right) y - \frac{top + bottom}{top - bottom}$$

z 축 좌표는 부호가 바꾸어 적용

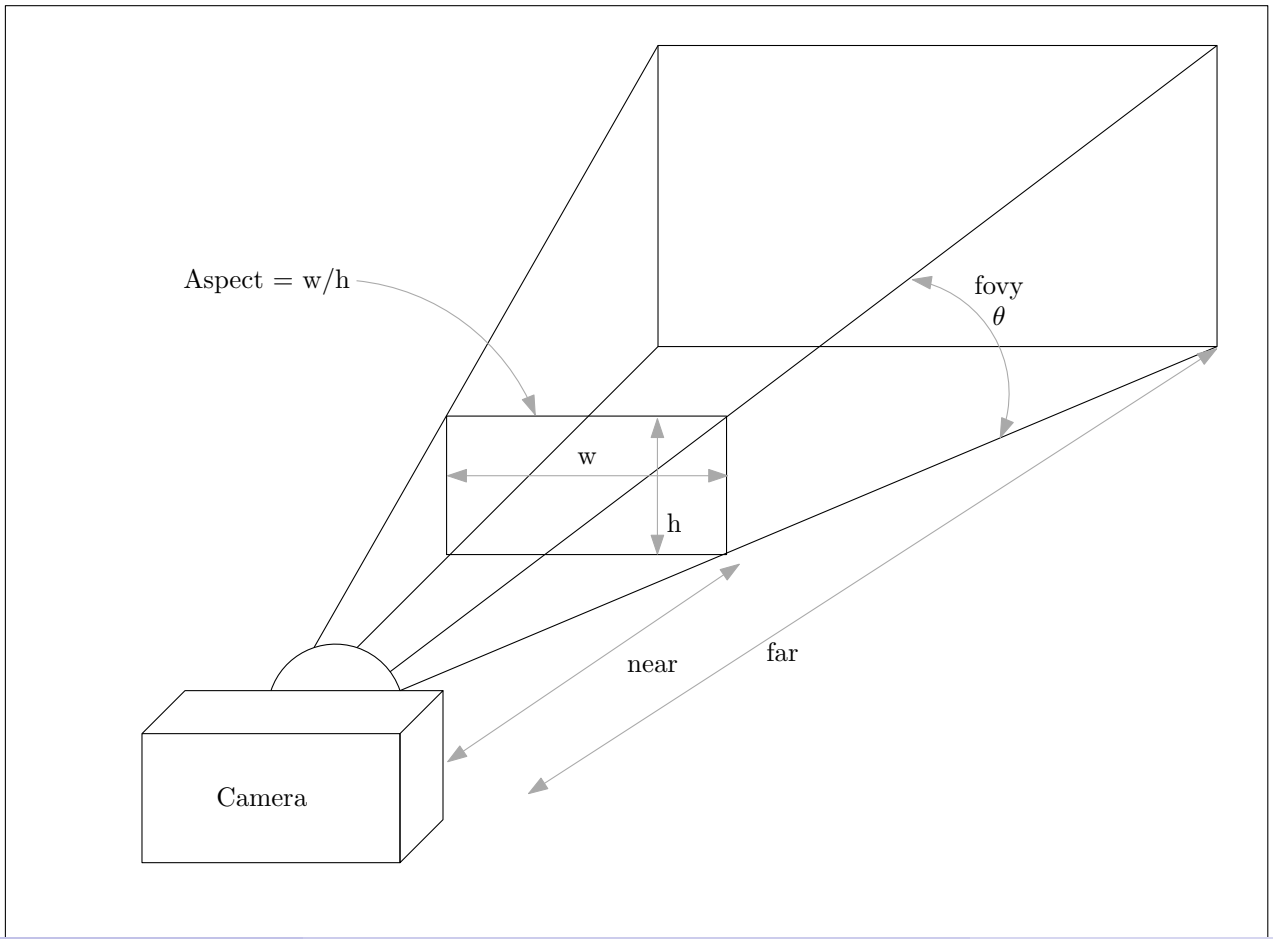
$$z' = -\left(\frac{2}{far - near}\right) z - \frac{far + near}{far - near}$$

glOrtho에 의해 결정되는 투영행렬

$$\begin{pmatrix} \frac{2}{right-left} & 0 & 0 & -\frac{right+left}{right-left} \\ 0 & \frac{2}{top-bottom} & 0 & -\frac{top+bottom}{top-bottom} \\ 0 & 0 & \frac{2}{far-near} & -\frac{far+near}{far-near} \\ 0 & 0 & 0 & 1 \end{pmatrix}$$

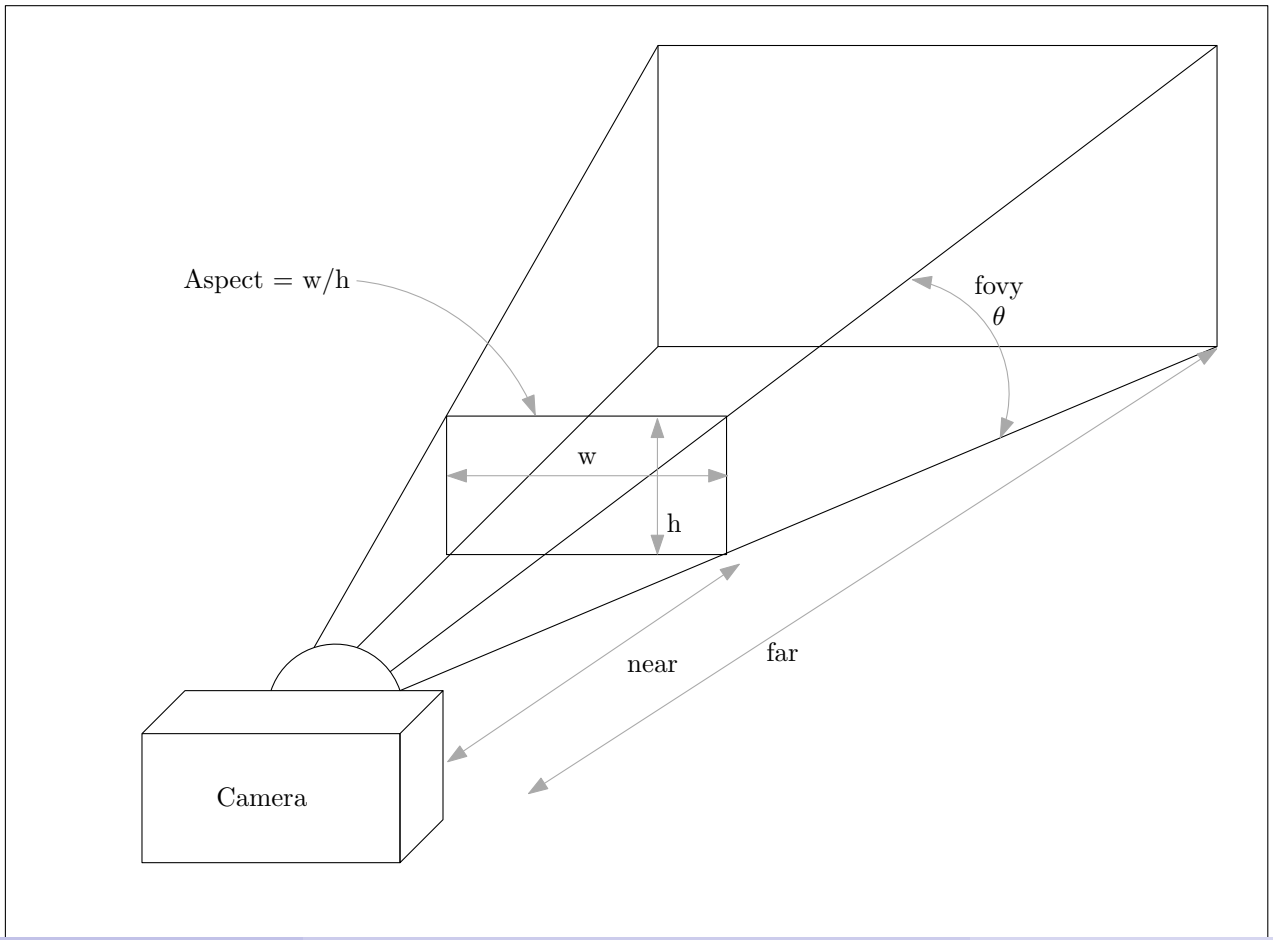
원근 투영 카메라

- 실제 카메라나 우리 눈
  - 멀리 있는 것은 작게 보이고 가까이 있는 것은 크게 보임
  - 빛이 관측 지점으로 모이면서 형성되는 선을 따라 원근투영되기 때문
- 이러한 원근 투영을 설정하는 방법은 glFrustum과 gluPerspective

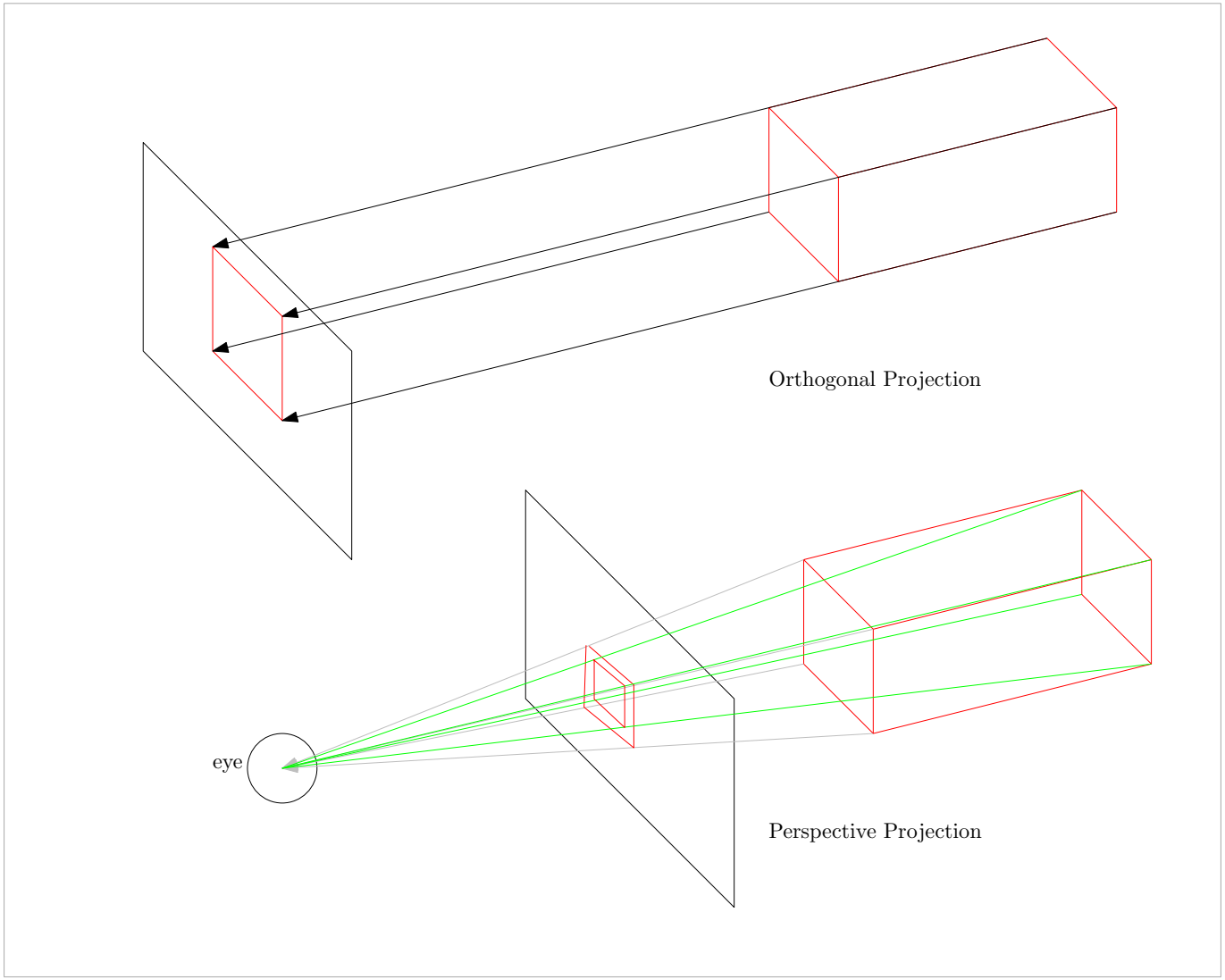


원근 투영 카메라

- 실제 카메라나 우리 눈
  - 멀리 있는 것은 작게 보이고 가까이 있는 것은 크게 보임
  - 빛이 관측 지점으로 모이면서 형성되는 선을 따라 원근투영되기 때문
- 이러한 원근 투영을 설정하는 방법은 glFrustum과 gluPerspective

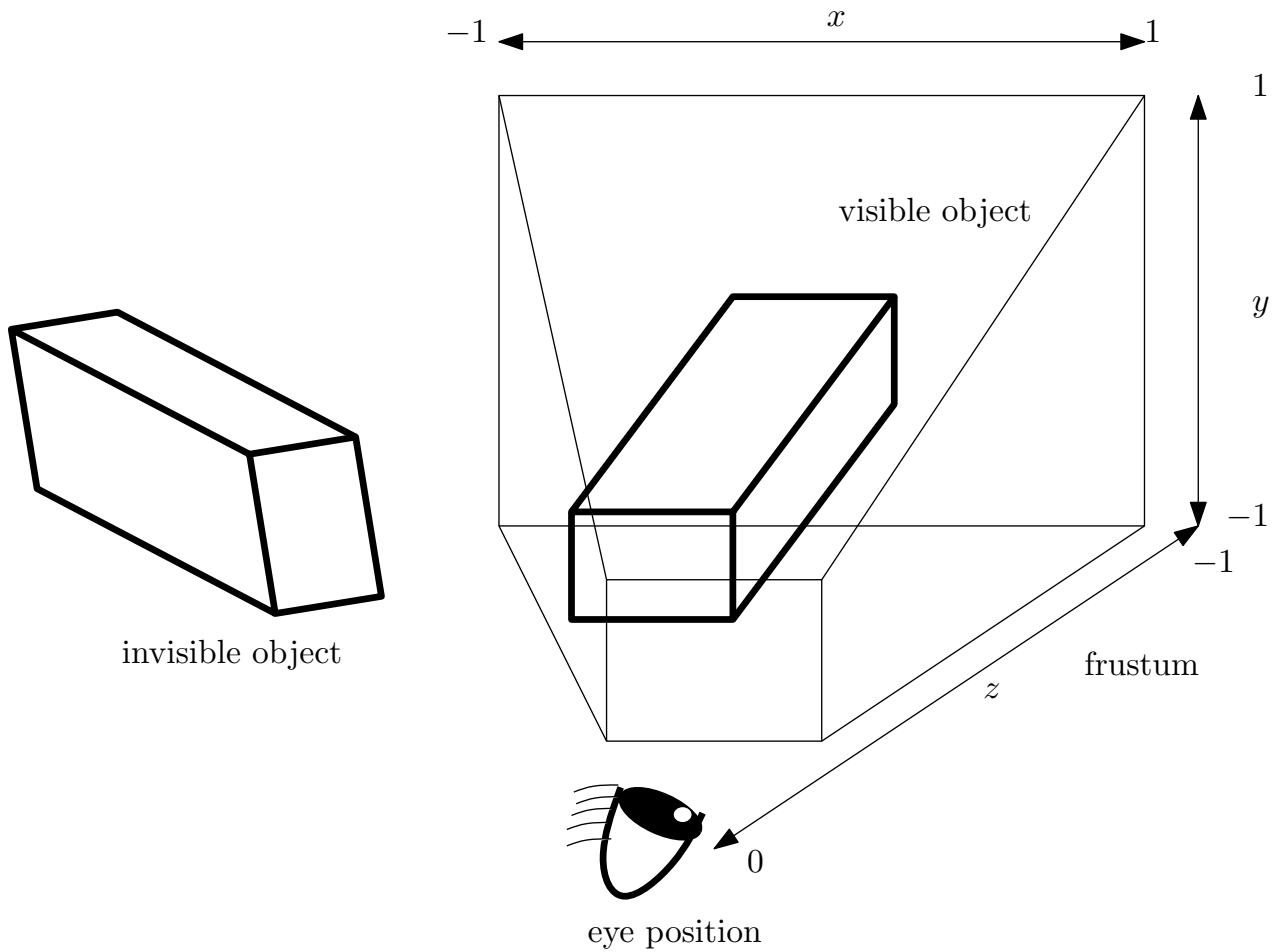


직교 투영과 원근 투영 비교



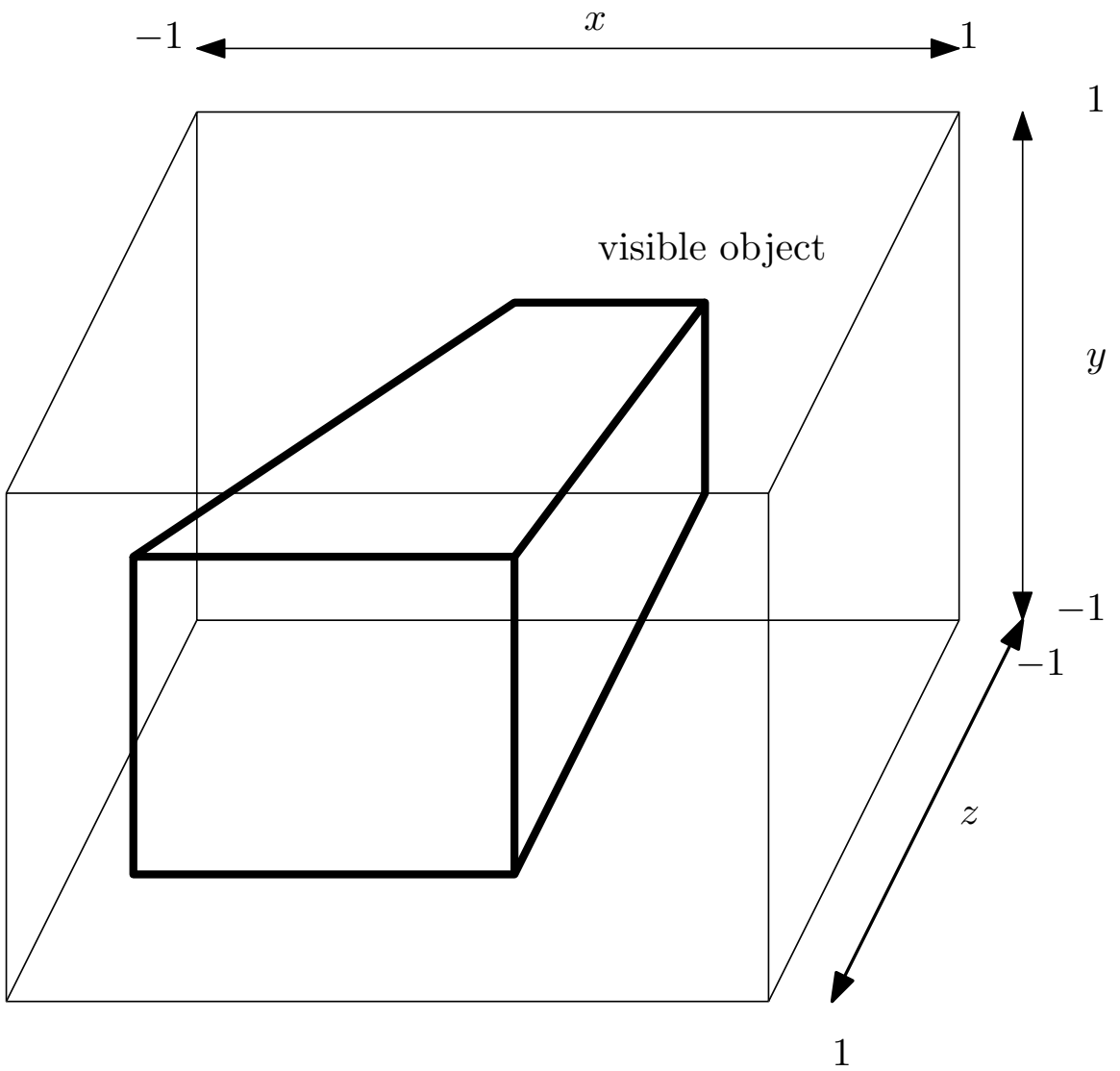
절두체 공간

- 3차원 그래픽스의 표준적인 카메라 모델은 렌더링의 대상이 되는 영역을 일정한 범위로 제한
- 원근이 있는 투영의 경우에는 절두체(frustum) 모양을 이룬다.



정규 장치 좌표계

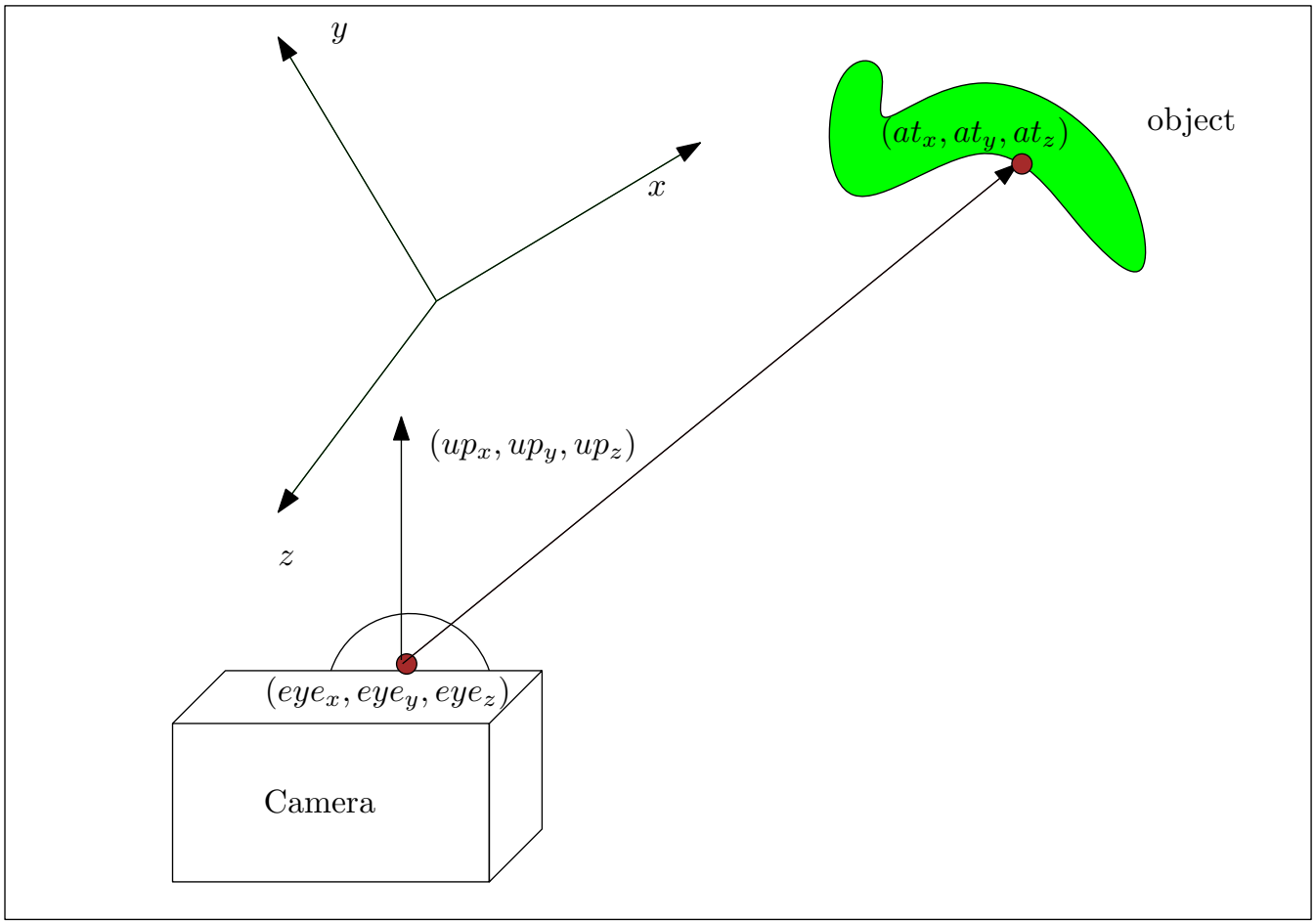
화면에 출력을 하기 위해서는 관측 볼륨을 정규 장치 좌표계로 바꾼다.



카메라의 위치변경

- gluPerspective를 사용하여도 카메라는 여전히 원점 (0,0,0)에 놓임
- 카메라를 원하는 곳으로 옮겨주는 함수: gluLookAt

gluLookAt(float eye_x, float eye_y, float eye_z, float at_x, float at_y, float at_z, float up_x, float up_y, float up_z)



행렬모드(Matrix mode)

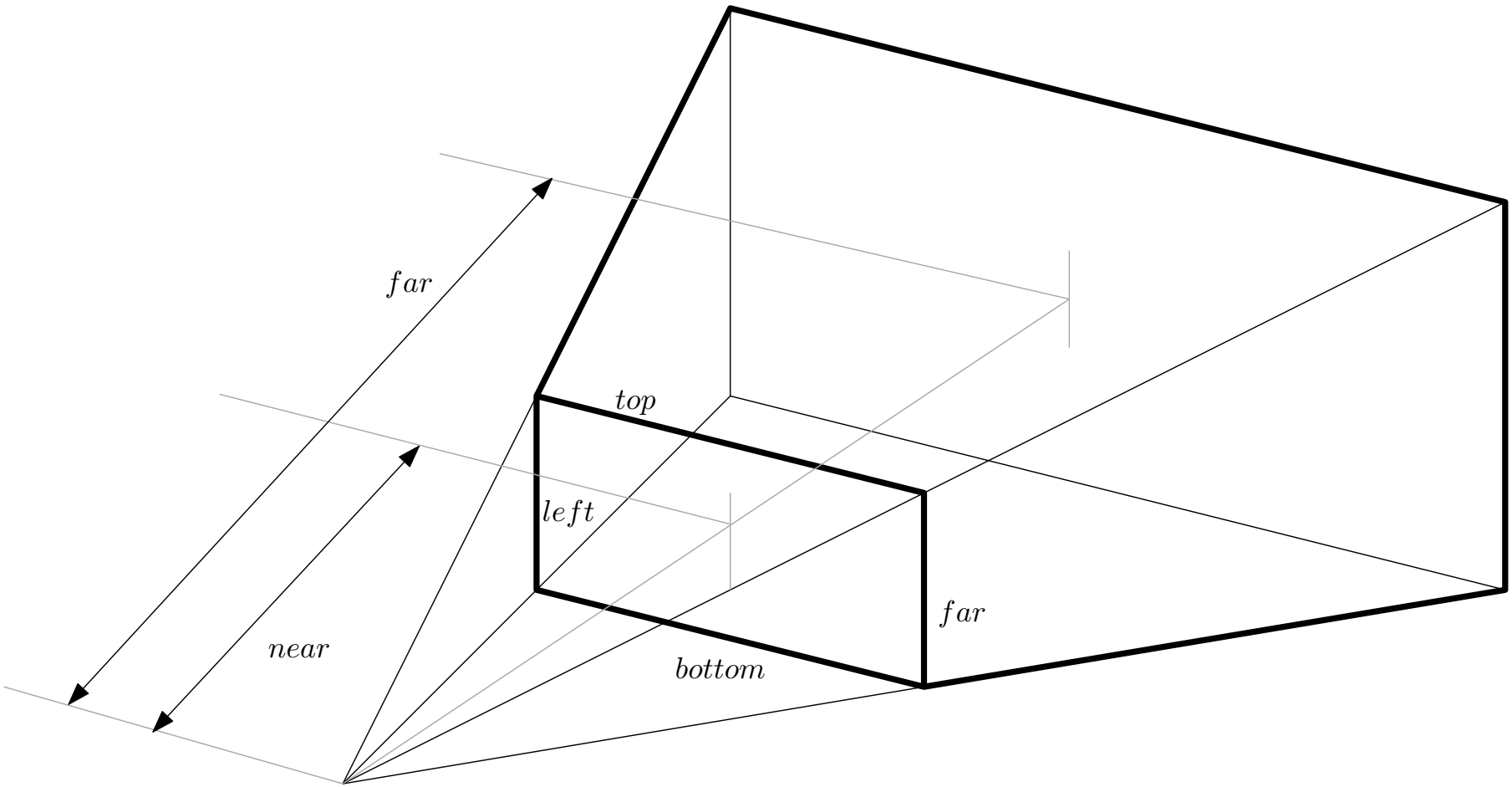
- OpenGL은 세 종류의 행렬 모드: 텍스처 행렬, 모델뷰 행렬, 투영행렬
- 가상공간 내의 좌표를 결정하는 행렬은 모델뷰 행렬과 투영행렬
- 모델뷰행렬(modelview matrix)은 공간 내에서 가상 객체를 변환하여 좌표를 변경하는 데에 사용
- 투영행렬(projection matrix)은 가상 객체를 투영면에 옮겨 놓는 데에 사용
- gluPerspective 함수는 투영의 특성을 변경하는 것이므로 투영행렬 모드
- gluLookAt은 투영의 특성이 아니라 카메라의 위치를 옮기는 것이고, 이는 바꾸어 말해 물체의 위치를 카메라 기준에서 옮기는 것이므로 모델뷰 행렬을 변경

```
glMatrixMode(GL_PROJECTION);
glLoadIdentity();
gluPerspective(60, 1.0, 0.1, 100.0);
glMatrixMode(GL_MODELVIEW);
glLoadIdentity();
gluLookAt(1.0,1.0,1.0, 0.0, 0.0, 0.0, 0.0, 1.0, 0.0);
```

glFrustum

카메라의 원근 투영을 설정하는 가장 기본적인 방법은 glFrustum 함수를 이용

glFrustum(float left, float right, float bottom, float top, float near, float far);



glFrustum

glFrustum 공간을 평행 투영 공간으로 바꾸기 위한 변환

$$\begin{pmatrix} \frac{2near}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & \frac{-2near \cdot far}{far-near} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

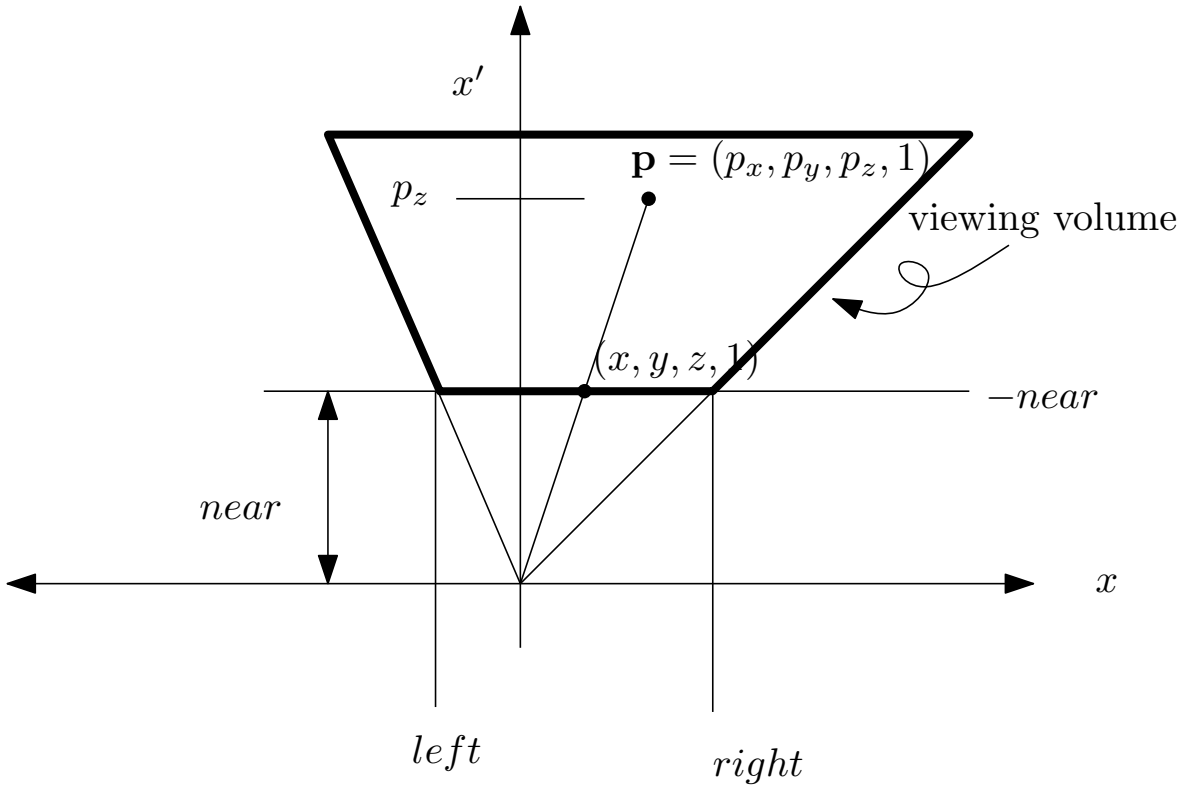
glFrustum

glFrustum 공간을 평행 투영 공간으로 바꾸기 위한 변환

$$\begin{pmatrix} \frac{2near}{right-left} & 0 & \frac{right+left}{right-left} & 0 \\ 0 & \frac{2near}{top-bottom} & \frac{top+bottom}{top-bottom} & 0 \\ 0 & 0 & -\frac{far+near}{far-near} & \frac{-2near \cdot far}{far-near} \\ 0 & 0 & -1 & 0 \end{pmatrix}$$

glFrustum 투영 유도

- 볼륨 안의 $\mathbf{p}$가 가까운 쪽 클리핑 평면 $z = -near$에 떨어진 좌표 구하기



$p_x : p_z = x : -near$임을 알 수 있고, 다음과 같은 식을 유도할 수 있다.

$$x = -near\frac{p_x}{p_z} \tag{1}$$

glFrustum 투영 유도

이렇게 해서 얻는 x 좌표를 직교 투영에서 살펴본 바와 같이 정규 장치 좌표계, 즉 $[-1, 1]$의 범위로 옮김

$$x' = \left(-\frac{2near}{right - left} \right) \frac{p_x}{p_z} - \frac{right + left}{right - left}]$$

glFrustum 투영 유도

비슷한 방법으로 y 좌표를 구하고 이를 정규 장치 좌표계의 좌표 y'로 바꾸는 작업

$$y' = \left(-\frac{2near}{top - bottom} \right) \frac{p_y}{p_z} - \frac{top + bottom}{top - bottom} \tag{2}$$

glFrustum 투영 유도

x와 y의 투영과 같은 꼴로 표현

$$z' = \frac{\alpha}{p_z} + \beta$$

$[near,\ far]$를 [-1,1]로

$$\begin{aligned} -1 &= \frac{\alpha}{near} + \beta \\ 1 &= \frac{\alpha}{far} + \beta \end{aligned} \tag{3}$$

이 이원일차연립방정식을 풀면 α와 β를 다음과 같이 구할 수 있다.

$$\alpha = \frac{near \cdot far}{far - near}, \quad \beta = \frac{far + near}{far - near}$$

glFrustum 투영 유도

$$\begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} -x \cdot p_x \\ -y \cdot p_y \\ -z \cdot p_z \\ -p_z \end{pmatrix} = \begin{pmatrix} \frac{2near}{right-left} p_x + \frac{right+left}{right-left} p_z \\ \frac{2near}{top-bottom} p_y + \frac{top+bottom}{top-bottom} p_z \\ -\frac{far+near}{right-left} p_z - \frac{2near \cdot far}{far-near} \\ -p_z \end{pmatrix}$$

이상의 결과를 통해 우리는 glFrustum에 의해 만들어지는 투영행렬이 식 1과 같음을 알 수 있다. 여기서 far의 값이 무한대인 경우, 즉, 먼쪽 클리핑 평면이 없는 경우의 투영행렬이 어떻게 될지 고민해 보라.

glFrustum과 gluPerspective

```
gluPerspective(fovy, aspect, near, far);
glFrustum(left,right,bottom,top,near,far);
```

$$top = near \cdot \tan \frac{\theta}{2}$$

$$bottom = -near \cdot \tan \frac{\theta}{2}$$

$$right = top \cdot aspect$$

$$left = bottom \cdot aspect$$