

7장. 투상변환과 뷰포트변환

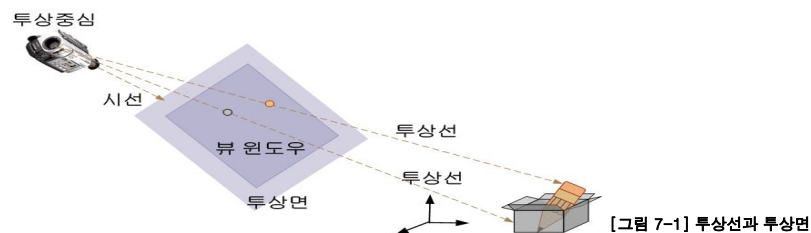
👤 학습목표

- 평행투상과 원근투상의 차이점을 이해한다.
- 가시부피 설정방식을 이해한다.
- `glOrtho()` 함수와 `gluPerspective()` 함수 파라미터를 이해한다.
- 전방 절단면을 되도록 시점에서 멀리 가져가는 이유를 이해한다.
- 시점좌표, 절단좌표, 정규화 장치좌표, 화면좌표로의 변환과정을 이해한다.

1

7.1 투상

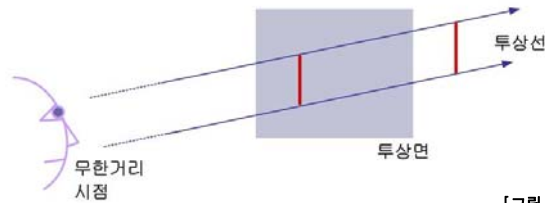
👤 투상(Projection) = 가시변환(Viewing Transformation)



- 투상면(View Plane, Projection Plane)
- 관찰자 위치(View Point, Eye Position)
= 카메라 위치(Camera Position) = **투상중심(COP: Center of Projection)**
= 시점좌표계 원점(Origin of VCS)
- 투상선(Projectors): 물체 곳곳을 향함
- 시선(Line of Sight): WCS원점 또는 초점을 향함
- 투상면(Projection Plane, View Plane)

2

평행투상(Parallel Projection)



[그림 7-2] 평행투상

👤 시점이 물체로부터 무한대의 거리에 있다고 간주

- 투상선이 평행
- 원래 물체의 평행선은 투상 후에도 평행
- 시점과의 거리에 무관하게 같은 길이의 물체는 같은 길이로 투상

👤 정사투상, 축측투상, 경사투상 등으로 분류

3

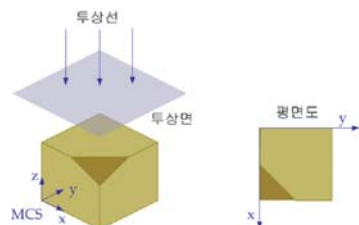
정사투상(Orthographic Projection)

👤 평면도, 입면도, 측면도 등

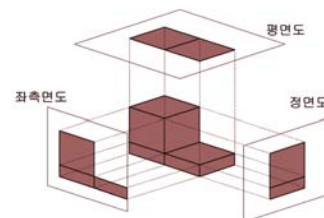
- 주 평면(Principal Plane): MCS 주축인 x, y, z 에 의해 형성되는 $x-y, y-z, z-x$
- 투상면은 주 평면 중 하나와 평행

👤 투상선은 투상면과 직교

- 원래 물체의 길이를 정확히 보존. 공학도면에 사용
- 투상선이 반드시 투상면과 직교-> 시점위치가 제한됨.



[그림 7-3] 정사투상 I



[그림 7-4] 정사투상 II

4

축측투상(Axonometric Projection)

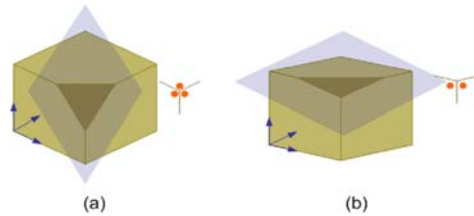
한꺼번에 여러 면을 보여줌

- 투상선은 투상면과 직교. 투상면이 주평면들과 평행하지 않음

축 방향으로 서로 다른 축소율(cf. 정사투상)

- 삼각(삼중형, Trimetric)
 - 투상면이 임의의 위치.
- 양각(이중형, Dimetric)
 - 투상면이 2 개의 주 평면에 대해서 대칭적.
 - 2개의 축 방향에 대해 동일 축소율
- 등각(동형, Isometric)
 - 투상면이 3 개의 주 평면이 만나는 모서리에서 모든 평면에 대해 대칭적으로 놓일 때. 3개의 축 방향에 대해 동일 축소율

Ex. 등각, 양각



[그림 7-5] 등각투상과 양각투상

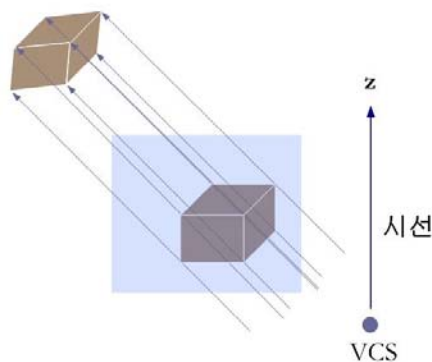
5

경사투상(Oblique Projection)

투상선끼리는 평행

투상면은 시선에 수직이지만 투상선과 직교하지 않음.

고개는 돌리지 않고 눈동자만 돌려서 보는 것과도 흡사

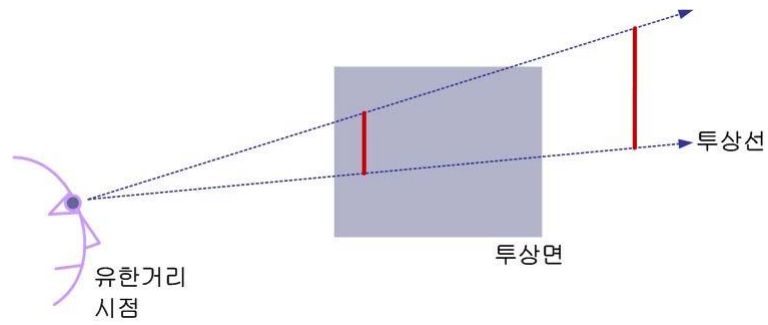


[그림 7-7] 경사투상

6

원근투상(Perspective Projection)

- 시점이 물체로부터 유한한 거리에 있다고 간주
- 투상선이 시점에서 출발하여 방사선 모양으로 퍼져감.
- 카메라나 사람의 눈이 물체를 포착하는 방법

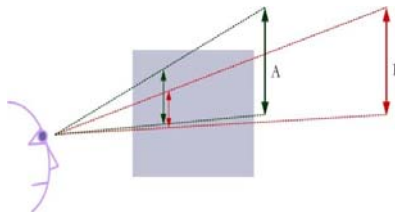


[그림 7-8] 원근투상

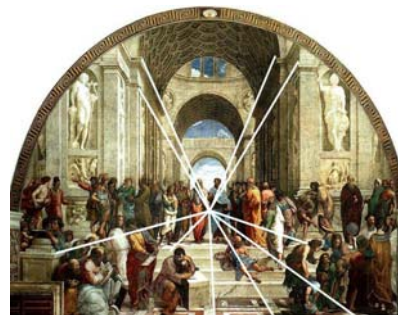
7

원근투상(Perspective Projection)

- 원근감(Depth Feeling)
 - 동일한 크기의 물체라도 시점으로부터 멀리 있는 것은 작게 보이고 가까운 것은 크게 보임



[그림 7-9] 축소율 차이



[그림 7-10] 라파엘, "아테네학당"

8

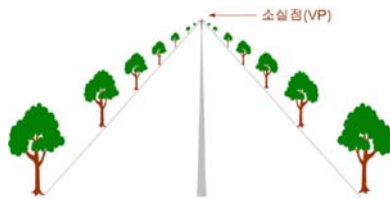
원근투상(Perspective Projection)

소실점(VP: Vanishing Point)

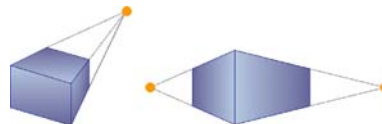
- 원근투상 결과 평행선이 만나는 점(시점 높이)
- 소실점의 수
 - 일점투상(One-point Projection), 이점투상(Two-point Projection), 삼점투상(Three-point Projection)

원근변환(Perspective Transformation)

- 직선->직선, 평면->평면
- 물체 정점간의 거리에 대한 축소율이 달라짐. (cf. 어파인 변환)



[그림 7-11] 소실점

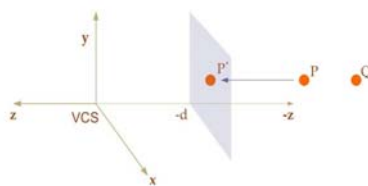


[그림 7-12] 일점투상과 이점투상

9

7.2 지엘의 투상변환-지엘의 평행투상

투상: `void glMatrixMode(GL_PROJECTION);`



$$P' = M_{\text{parallel}} \cdot P$$

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

[그림 7-13] 평행투상

(7.1) (7.2)

기본 평행투상

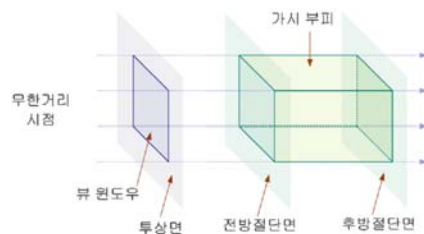
- 모델 좌표, 전역 좌표, 시점 좌표 순서로 변환된 상태
- P, P'은 시점 좌표계 기준의 좌표. 거리 d에 무관하게 동일한 모습
- 특이변환(Singular Transformation): 역변환이 없는 변환
 - (x, y, z, 1)에서 (x, y)만 읽어내면 그것이 투상된 2차원 좌표
 - 깊이 정보를 활용하기 위해서 지엘은 이러한 변환을 가하지 않음
 - 투상결과 여전히 3차원 좌표가 유지.

10

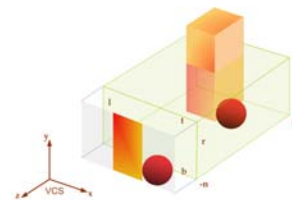
가시부피에 의한 평행투상

장면의 범위를 지정할 필요성: 가시부피(View Volume)

- 전방 절단면(Near Clipping Plane, Near Plane, Front Plane, Hither)
- 후방 절단면(Far Clipping Plane, Far Plane, Back Plane, Yon)



[그림 7-14] 평행투상의 가시부피



[그림 7-15] 절단

- `void glOrtho(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);`

11

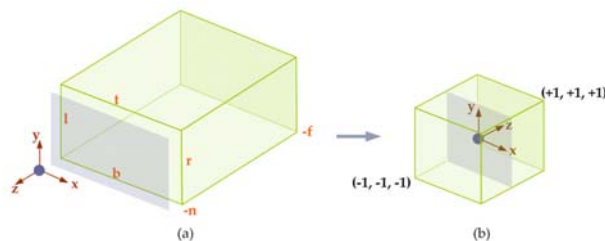
정규화 가시부피

정규화 가시부피(CVV: Canonical View Volume)

- 가로, 세로, 높이가 2인 정육면체로 투상
- 정규화 변환(Normalization Transformation)

이유

- 평행투상, 원근투상을 동일한 모습의 정규화 가시부피로 변형
 - 동일 파이프라인 사용
- 정육면체를 기준으로 하면 연산이 간단함.
- 다양한 해상도의 화면 좌표계로 변환하기가 간단함.

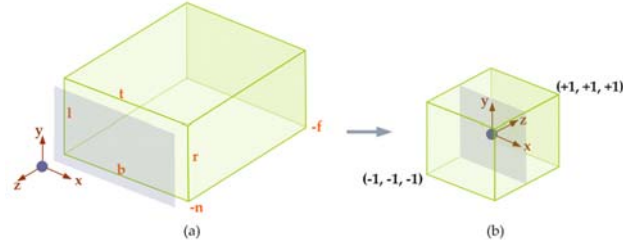


[그림 7-16] 평행투상에서의 가시부피 정규화

12

정규화 가시부피 변환

[그림 7-16] 평행투상에서의 가시부피 정규화



$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0 & 1 & 0 & -\frac{t+b}{2} \\ 0 & 0 & 1 & \frac{f+n}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad S = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad (7.7)$$

$$N = R \cdot S \cdot T$$

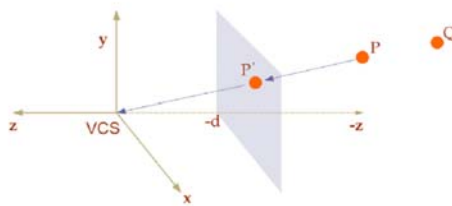
- 물체에 대한 이동, 크기조절, 반사변환으로 간주
- Reflection:** 정규화 가시부피는 왼손좌표계
- 결과적인 좌표계 = **절단 좌표계(CCS: Clip Coordinate System)**

13

지엘의 원근투상

기본 원근투상

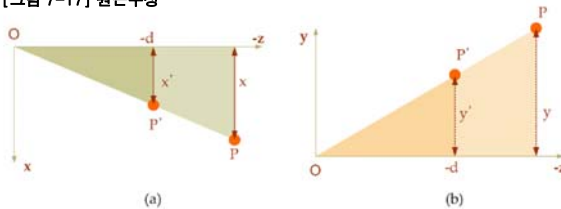
- $x' : (-d) = x : (-z), y' : (-d) = y : (-z)$



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

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

[그림 7-17] 원근투상



$$= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (7.8) \quad (7.9) \quad (7.10)$$

[그림 7-18] 평면도와 측면도

14

원근변환

원근분할(Perspective Division, Homogenization)

- 등차좌표의 마지막 요소로 이전 요소를 나누는 작업
- 절단이 등차좌표에서 이루어지기 때문에 절단 이후로 미루어 짐.

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

원근변환

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

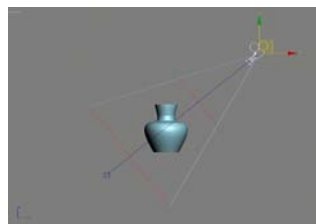
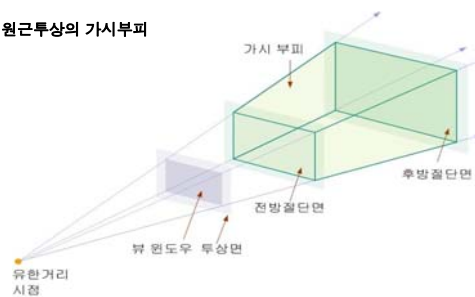
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = M_{\text{perspective}} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

15

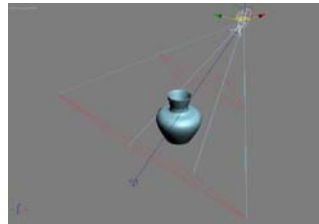
정규화 가시부피에 의한 원근투상

절단 사각뿔(Frustum) = 절두체

[그림 7-19] 원근투상의 가시부피



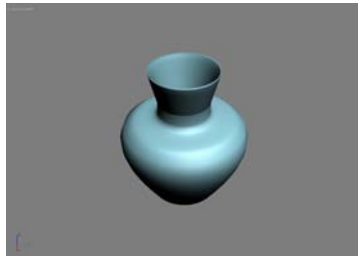
[그림 7-20] 가시부피 예시 I



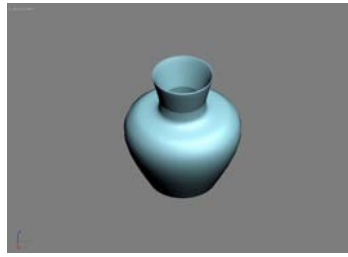
[그림 7-21] 가시부피 예시 II

16

원근투상과 평행투상



[그림 7-22] 원근투상 결과

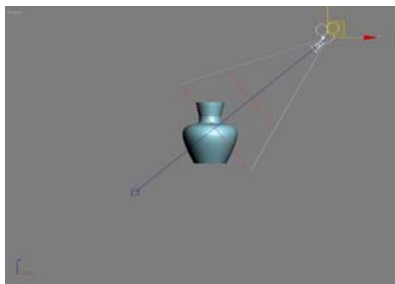


[그림 7-23] 평행투상 결과

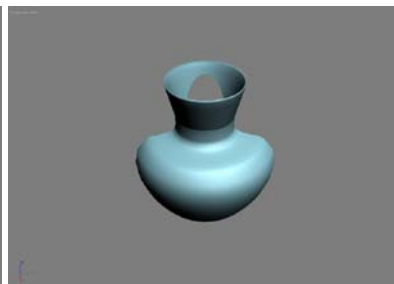
17

가시부피

 가시부피 설정에 의한 절단



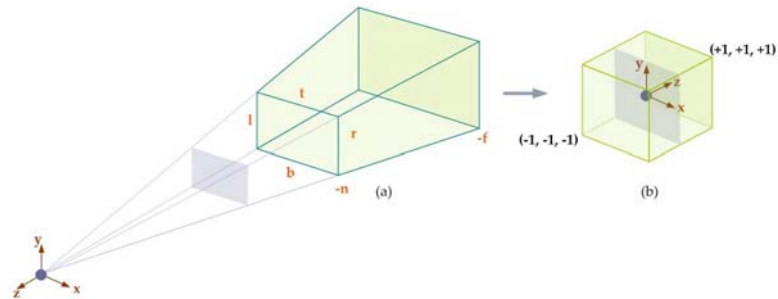
[그림 7-24] 가시부피 설정



[그림 7-25] 절단된 물체

18

원근투상의 정규화 가시부피



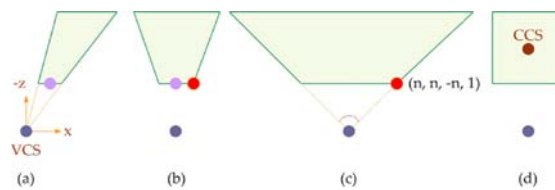
[그림 7-26] 원근투상에서의 가시부피 정규화

19

원근투상의 정규화 가시부피

일반적 형태의 가시부피

- `void glFrustum(GLdouble left, GLdouble right, GLdouble bottom, GLdouble top, GLdouble near, GLdouble far);`



[그림 7-27] 정규화 변환

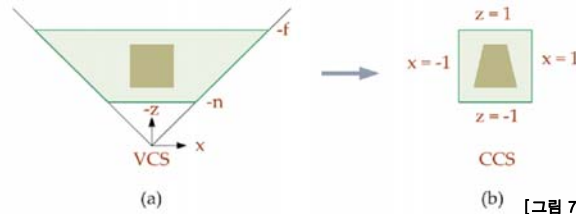
$$\begin{aligned}
 (7.15) \quad N &= T \cdot S \cdot Sh \\
 &= \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & \frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} \frac{2n}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2n}{t-b} & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} 1 & 0 & \frac{r+l}{2n} & 0 \\ 0 & 1 & \frac{t+b}{2n} & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \\
 &= \begin{pmatrix} \frac{2n}{r-l} & 0 & \frac{r+l}{r-l} & 0 \\ 0 & \frac{2n}{t-b} & \frac{t+b}{t-b} & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix}
 \end{aligned}$$

20

원근투상의 정규화 가시부피

• 전방 절단면에 비해 후방 절단면이 줄어듬.

- 멀리 있는 것이 작게 보여야 함.



[그림 7-29] 정규화 변환 결과

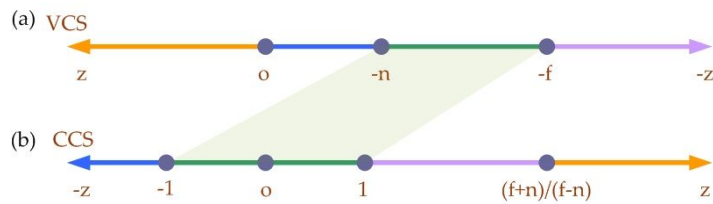
• T 행렬

- Z 값에 영향을 미침: 원래의 물체 정점의 깊이 z 와 정규화 변환 후의 물체 정점의 깊이 z' 의 관계

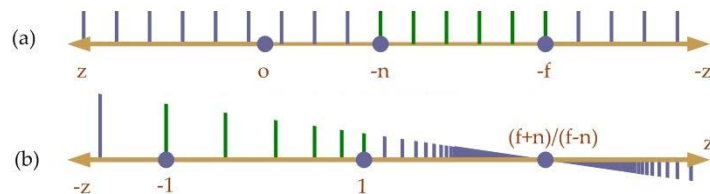
$$P' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -\frac{f+n}{f-n} & -\frac{2fn}{f-n} \\ 0 & 0 & -1 & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (7.17)$$

21

시점 좌표계에서 절단 좌표계로



[그림 7-30] 시점좌표계에서 절단좌표계로의 사상



[그림 7-31] 원근변환에 따른 간격 변화

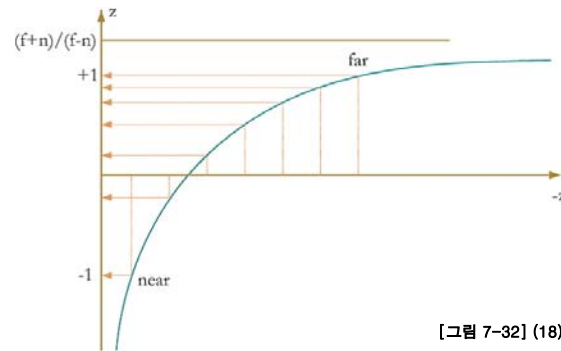
• 높이: 멀어질수록 전봇대 높이가 낮아짐(원근변환)

22

시점 좌표계에서 절단 좌표계로

간격: 더욱 촘촘해 짐에 유의(비선형 변환)

$$z' = \frac{f+n}{f-n} - \frac{2fn}{f-n} \frac{1}{(-z)} \quad (7.19)$$



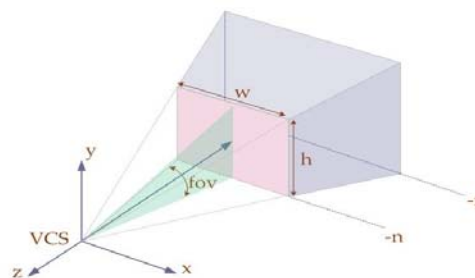
[그림 7-32] (18)식의 함수변화

- 전방 절단면: 시점에서 멀리, 물체에 최대한 근접시켜 설정
 - 물체간격이 상대적으로 보존, 지-버퍼 처리에 유리

23

대칭적 원근투상

void gluPerspective (GLdouble fovy, GLdouble aspect, GLdouble near, GLdouble far);



[그림 7-34] 시야각

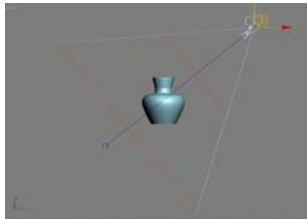
- 시야(FOV: Field of View): 상하 y축 방향의 시야각(0 -180도)
- X축 방향의 시야는 종횡비(Aspect Ratio)에 의해 결정됨(폭 / 높이)

24

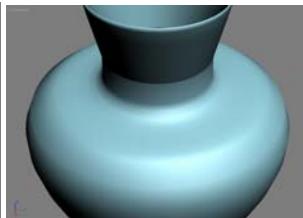
시야각과 카메라 렌즈

초점 거리 50mm 기준

- 광각렌즈(Wide Angle Lens) : 50보다 작음
- 망원렌즈(Telescope Lens) : 50보다 큼
- Ex. 20mm = 85도 시야각, 85mm = 24도 시야각



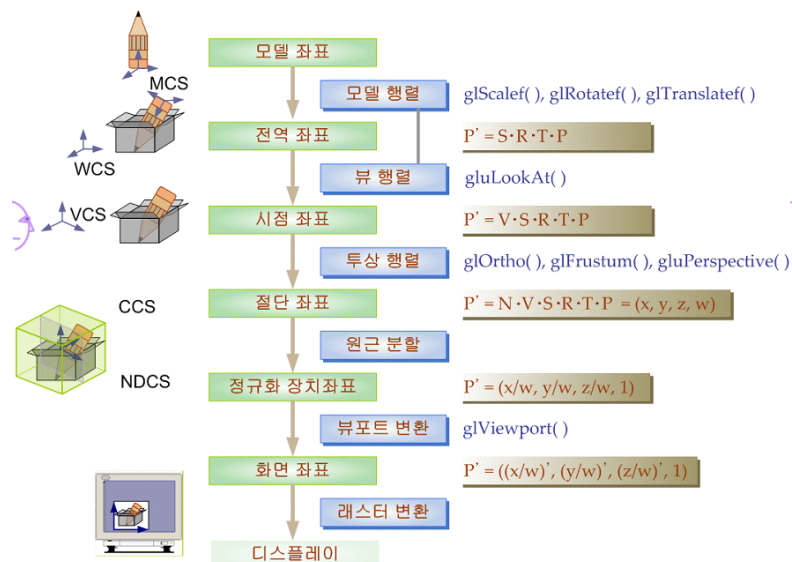
[그림 7-35/36] 20mm, 85도, 결과 I



[그림 7-37/37] 85mm, 24도, 결과 II

25

지엘 파이프라인



[그림 7-39] 지엘의 좌표변환 파이프라인

26

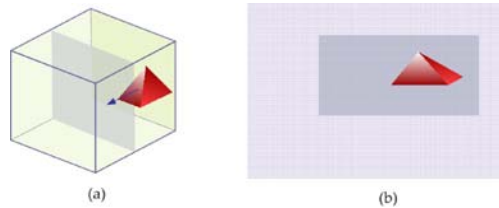
7.3 지엘의 뷰포트 변환-뷰포트 변환

정규화 장치좌표계(NDCS: Normalized Device Coordinate System)

- 절단 이후 원근분할에 의해 물체 정점을 3차원 좌표로 변환한 것
- $(x', y', z', 1) = (x/w, y/w, z/w, 1)$

뷰포트 변환(Viewport Transformation)

- 정규화 장치좌표계에서 화면 좌표계로 가는 작업
- 화면 좌표계(SCS: Screen Coordinate System), 뷰포트 좌표계(Viewport Coordinate System), 윈도우 좌표계(Window Coordinate System)

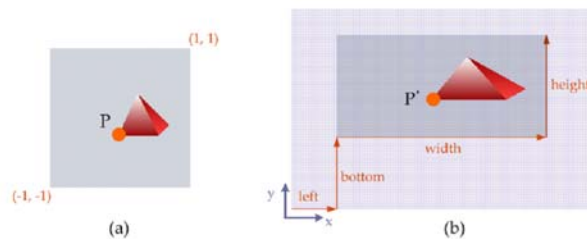


[그림 7-40] 뷰포트 변환

27

뷰포트 설정

`void glViewport(GLint left, GLint bottom, GLsizei width, GLsizei height);`



[그림 7-41] 뷰포트 설정

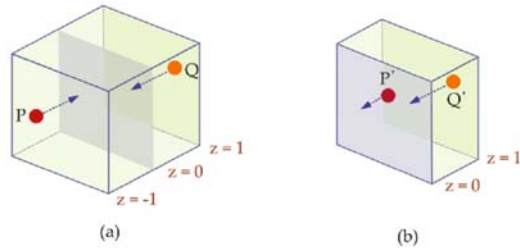
$$x' = \frac{x - (-1.0)}{(1.0) - (-1.0)} \times width + left \quad (7.22)$$

$$y' = \frac{y - (-1.0)}{(1.0) - (-1.0)} \times height + bottom \quad (7.23)$$

$$P' = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} \frac{width}{2} & 0 & 0 & left + \frac{width}{2} \\ 0 & \frac{height}{2} & 0 & bottom + \frac{height}{2} \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} \quad (7.24)$$

28

z 값의 재 정규화



[그림 7-42] 재 정규화

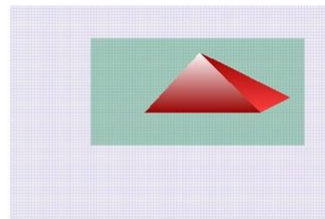
정규화 가시부피에서의 z 값의 범위인 $[-1, +1]$ 사이를 $[0, 1]$ 사이로 사상

- 점 사이의 상대적인 깊이는 유지
- 정규화 가시부피를 원점을 중심으로 해서 z 축 방향으로 $1/2$ 만큼 크기조절 변환을 가한 후, z 방향으로 $1/2$ 만큼 이동

29

가시부피와 뷰포트

(a)

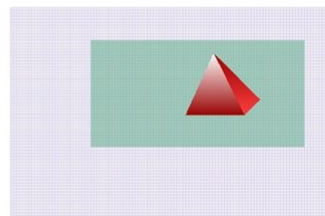


가시부피

정규화 가시부피
투상

뷰포트

(b)



[그림 7-43] 가시부피, 정규화 가시부피, 뷰포트의 관계

30