

# Part2: Rendering

## 1. 투상변환과 뷰포트 변환

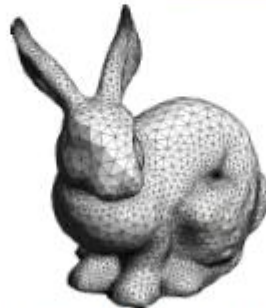
# Outline

- I. Rendering
- II. 투상 변환
  - A. Orthographic Projection
  - B. Perspective Projection
- III. 구현

# 1. Rendering



© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/help/faq-fair-use/>.



© Oscar Meruvia-Pastor, Daniel Rypl. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/help/faq-fair-use/>.

## Scene

# 1. Rendering



© source unknown. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/help/faq-fair-use/>.



© Oscar Meruvia-Pastor, Daniel Rypl. All rights reserved. This content is excluded from our Creative Commons license. For more information, see <http://ocw.mit.edu/help/faq-fair-use/>.

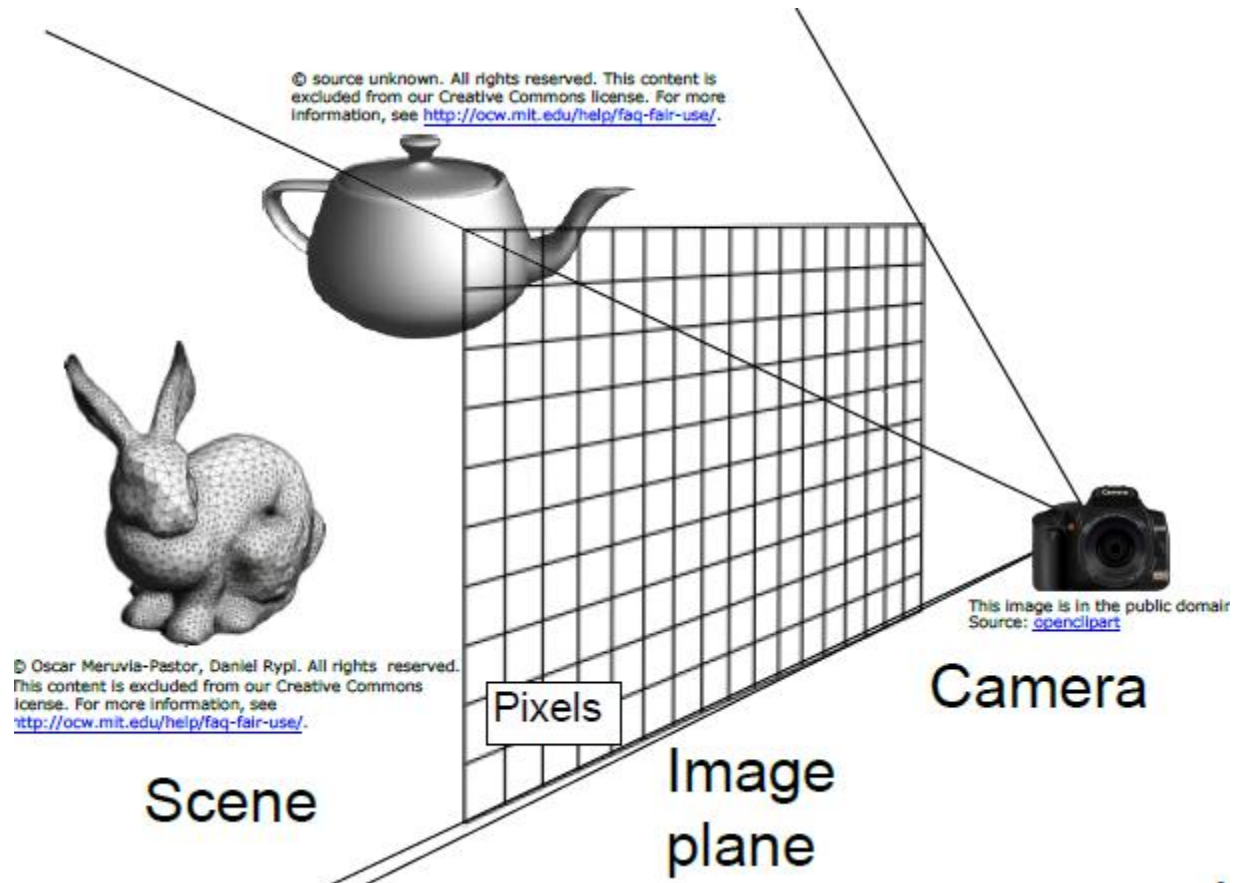
Scene



This image is in the public domain.  
Source: [openclipart](https://openclipart.org/)

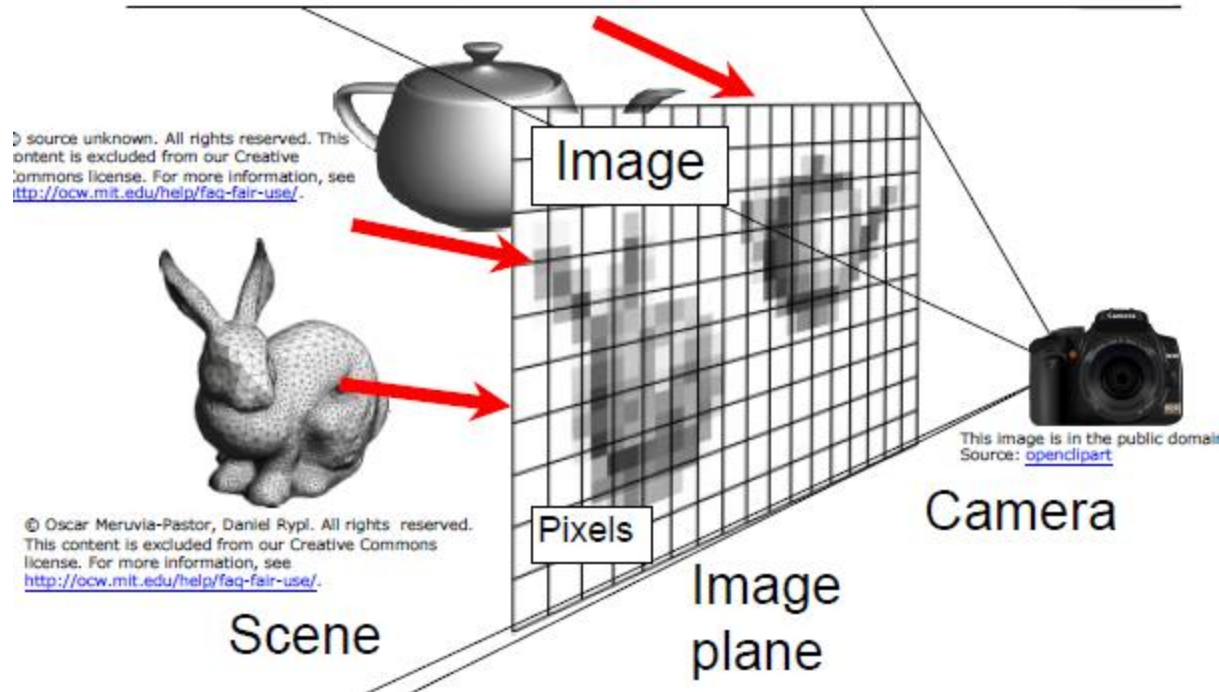
Camera

# 1. Rendering



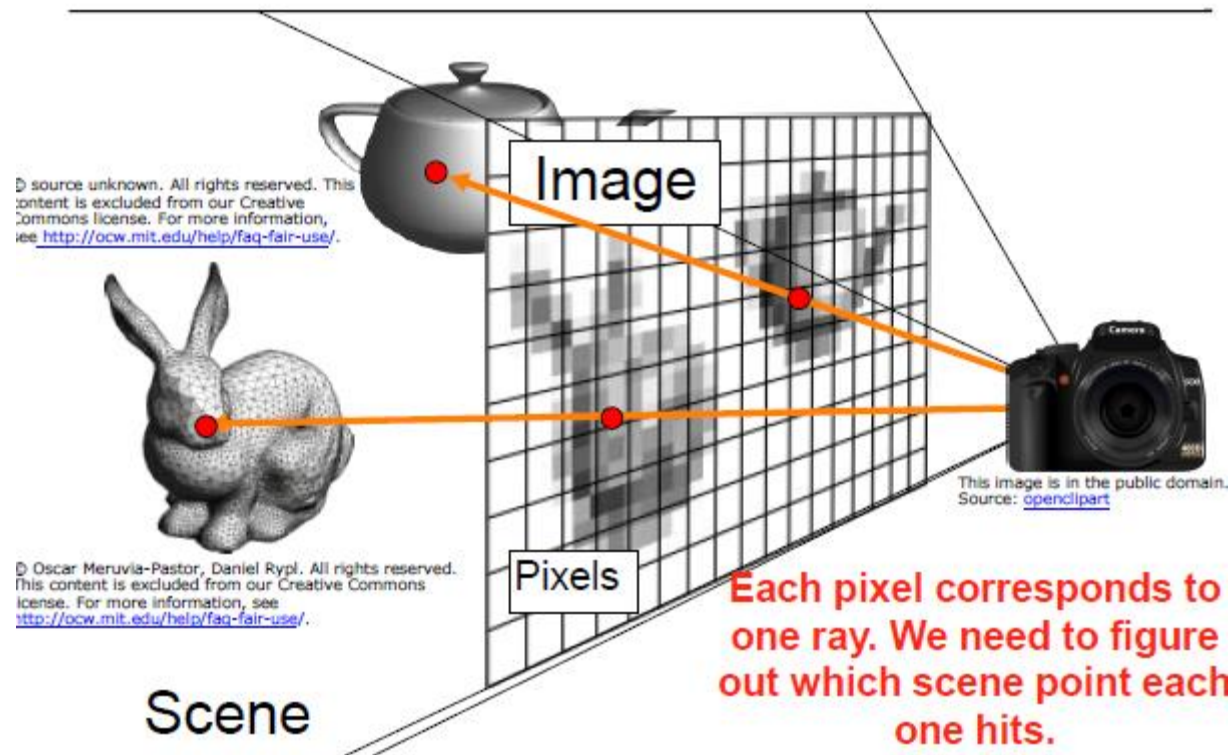
# 1. Rendering

Rendering = Scene to Image



# 1. Rendering

## Rendering – Pinhole Camera



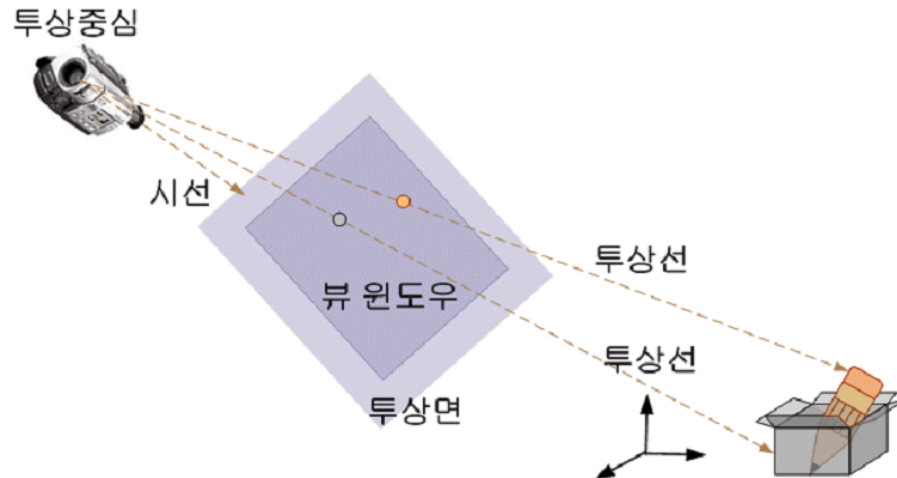
# 1. Rendering

- “Rendering” refers to the entire process that produces color values for pixels, given a 3D representation of the scene
- Pixels correspond to rays; need to figure out the **visible** scene point along each ray
  - Called “hidden surface problem” in older texts
  - “Visibility” is a more modern term
  - Also, we assume (for now) a single ray per pixel



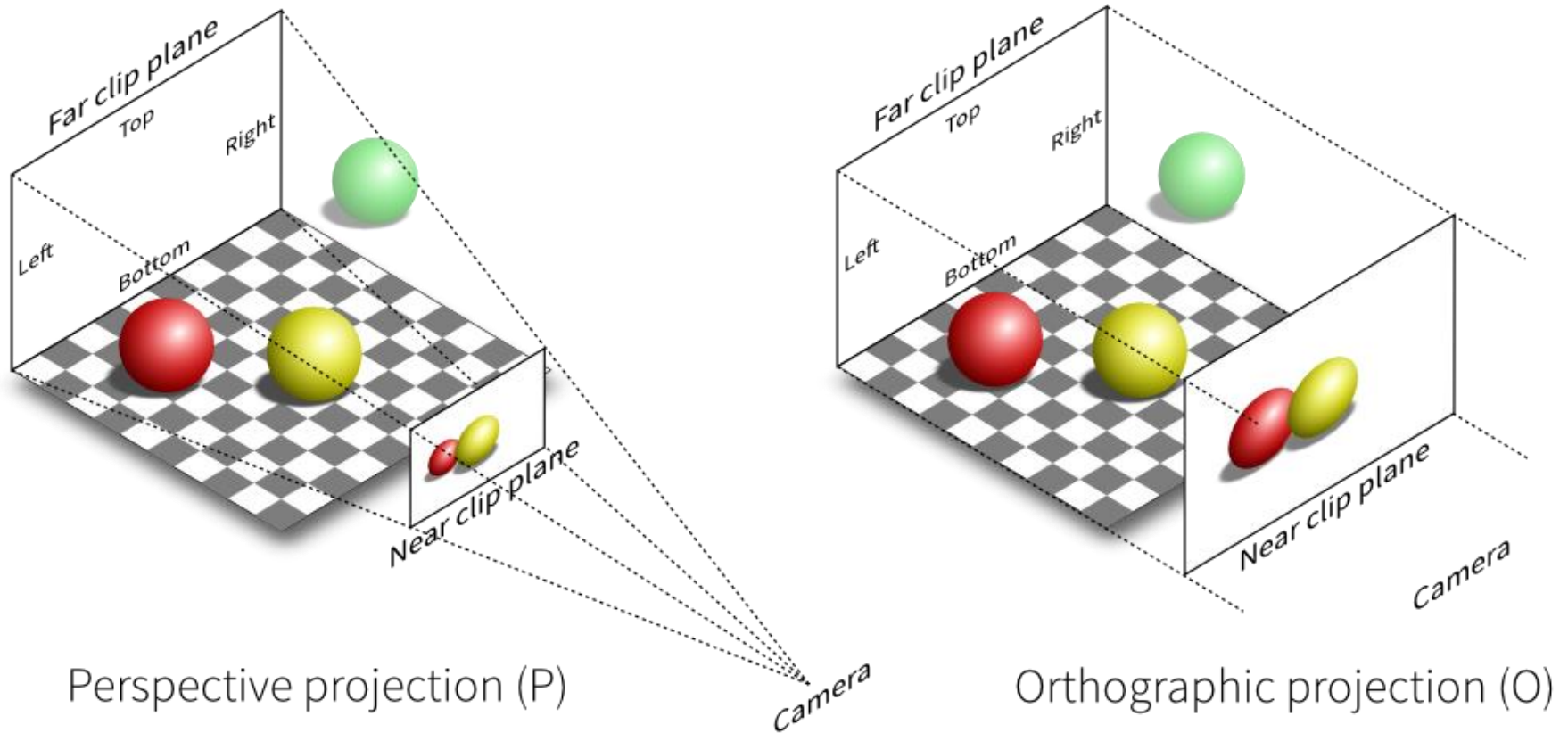
## 2. 투상변환

- 투상(Projection) = 가시 변환(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 초점 향함
- 투상면(Projection Plane, View Plane)



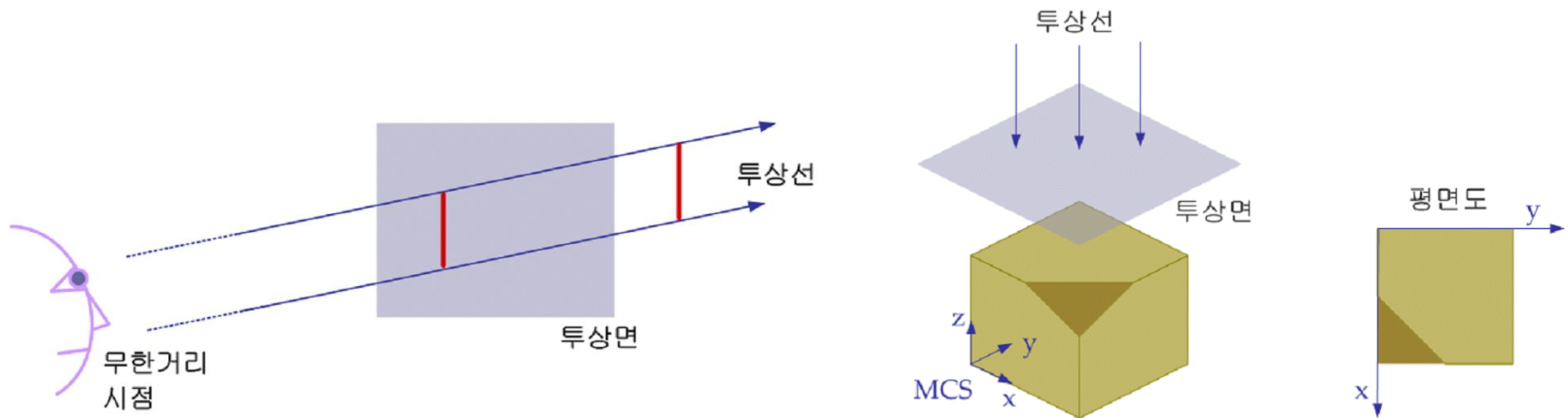
## 2. 투상변환

- Perspective Projection vs Orthographic Projection



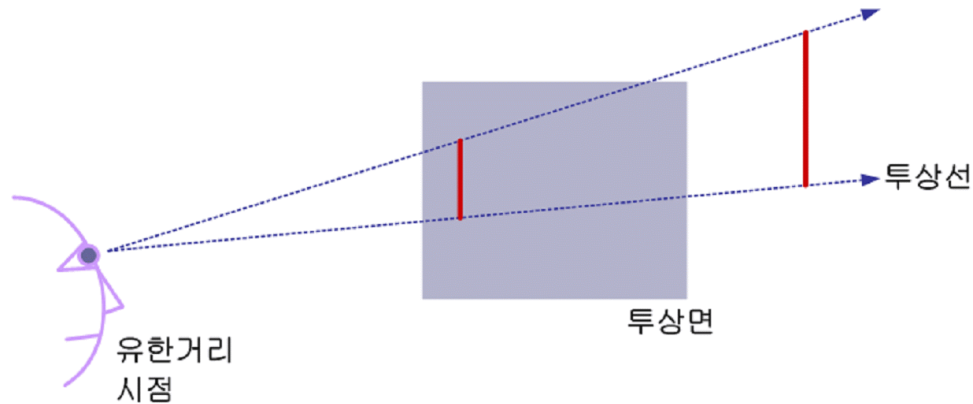
## 2.1 Orthographic Projection

- 시점이 물체로부터 무한대의 거리에 있다고 간주
  - 투상선이 평행
  - 원래 물체의 평행선은 투상 후에도 평행
  - 시점과의 거리에 무관하게 같은 길이의 물체는 같은 길이로 투상



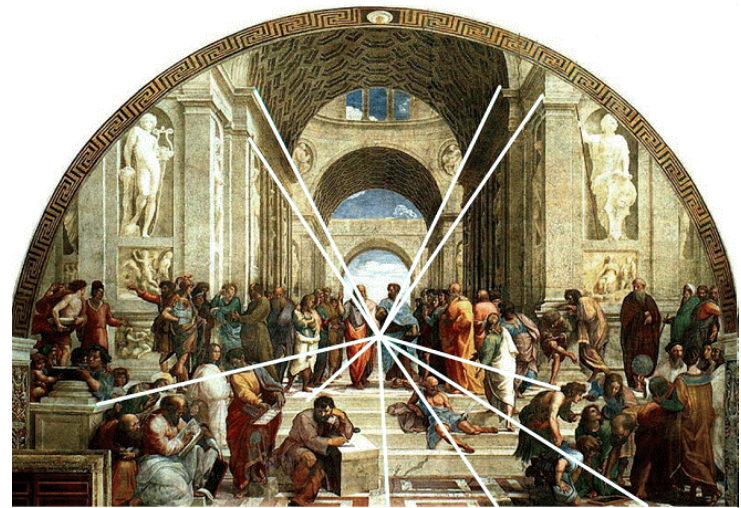
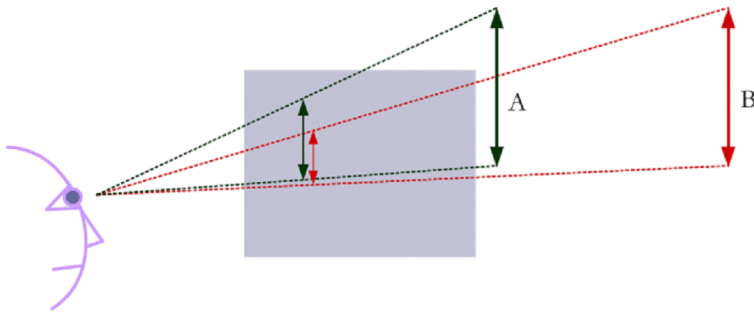
## 2.2 Perspective Projection

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



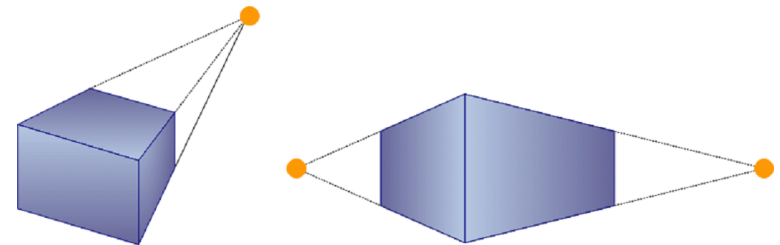
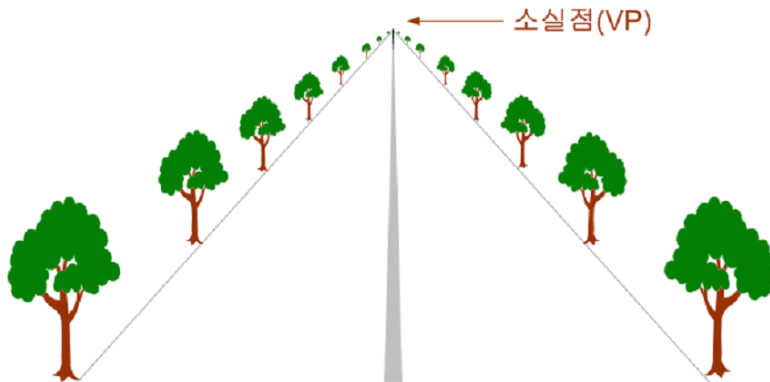
## 2.2 Perspective Projection

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

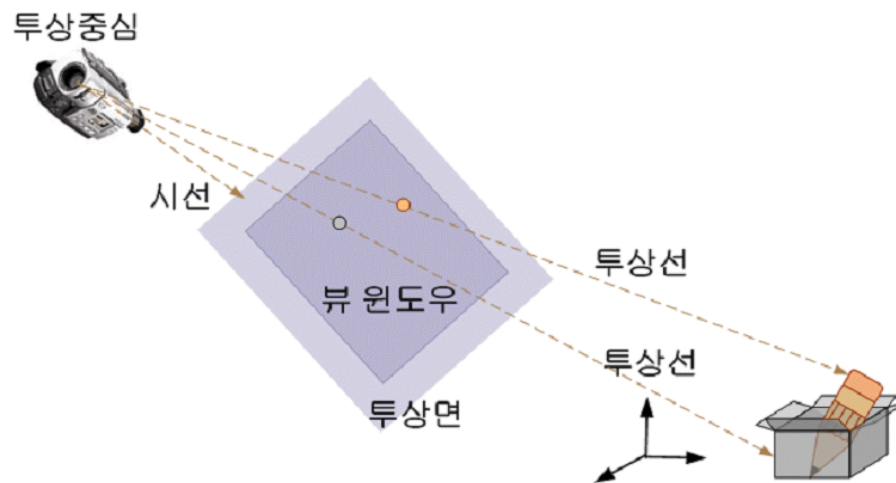


## 2.2 Perspective Projection

- 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)



### 3. 구현

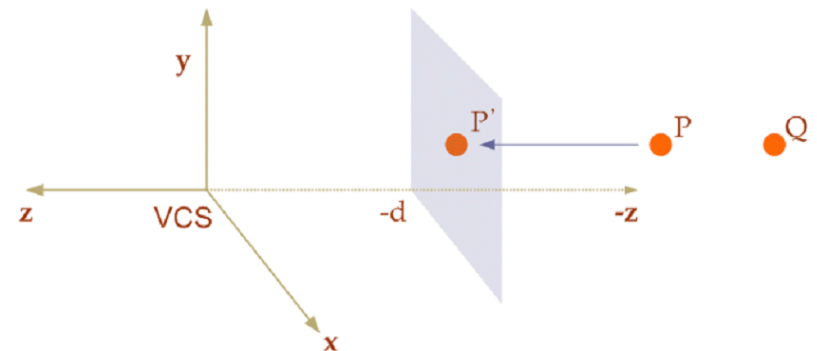


# 3.1 Orthographic

- 기본 평행투상
  - 모델 좌표, 전역 좌표, 시점 좌표 순서로 변환된 상태
  - $P, P'$ 은 시점 좌표계 기준의 좌표. 거리  $d$ 에 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.

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

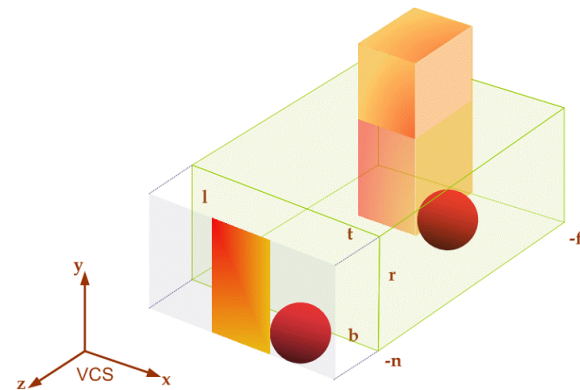
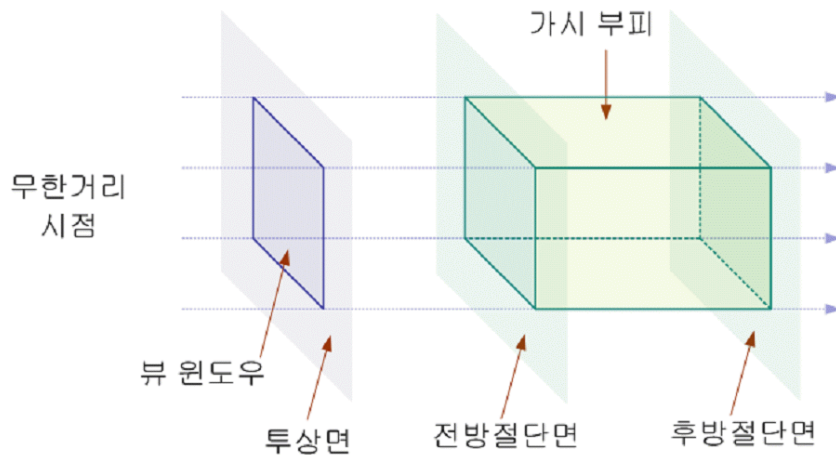
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & -d \\ 0 & 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x \\ y \\ -d \\ 1 \end{pmatrix}$$





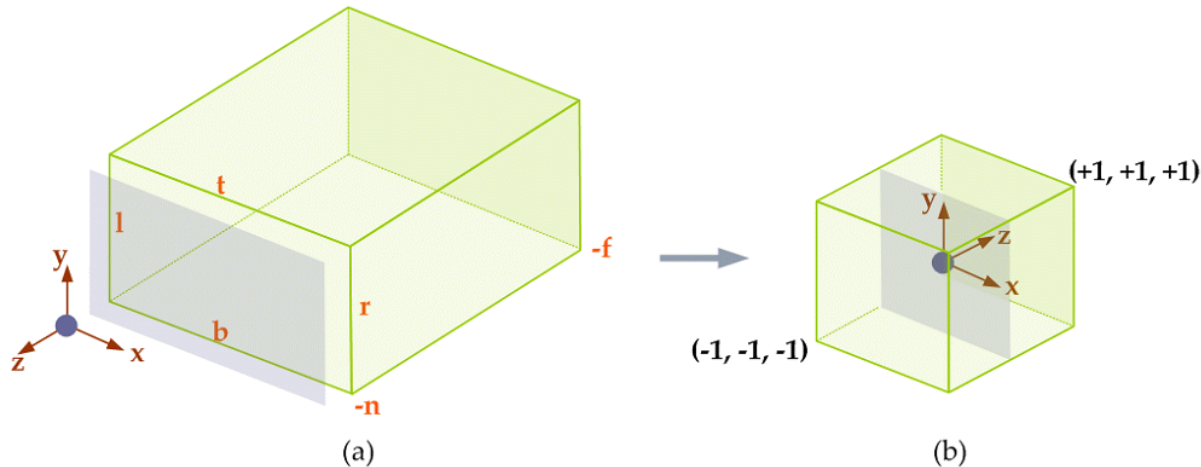
# 3.1.1 가시부피에 의한 평행투상

- 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)



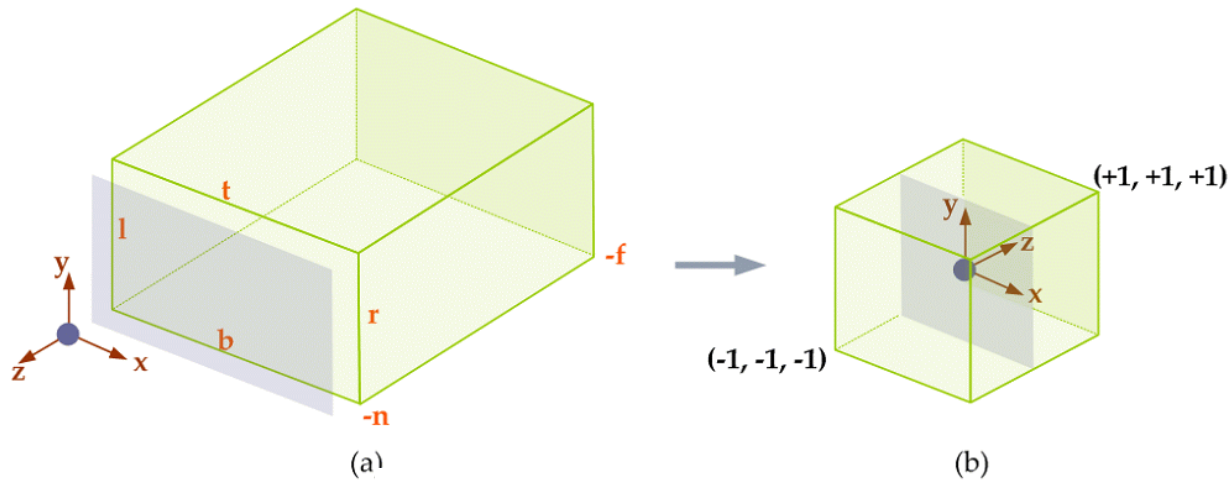
### 3.1.3 정규화 가시부피 변환

- 정규화 가시부피(CVV: Canonical View Volume)
  - 가로, 세로, 높이가 2인 정육면체로 투상
  - 정규화 변환(Normalization Transformation)
- 평행투상, 원근투상을 동일한 모습의 정규화 가시부피로 변형 → 동일 파이프라인 사용
- 정육면체를 기준으로 하면 연산이 간단함.
- 다양한 해상도의 화면 좌표계로 변환하기가 간단함.



### 3.1.3 정규화 가시부피 변환

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

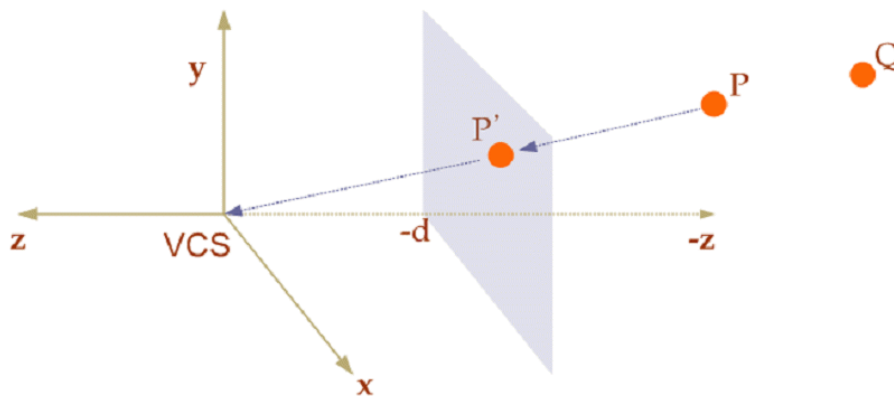


$$R = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad S = \begin{pmatrix} \frac{2}{r-l} & 0 & 0 & 0 \\ 0 & \frac{2}{t-b} & 0 & 0 \\ 0 & 0 & \frac{2}{f-n} & 0 \\ 0 & 0 & 0 & 1 \end{pmatrix} \quad T = \begin{pmatrix} 1 & 0 & 0 & -\frac{r+l}{2} \\ 0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

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

# 3.2 Perspective

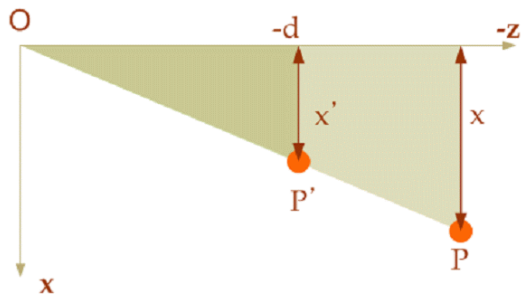
- 기본 원근투상
  - $x' : (-d) = x : (-z), y' : (-d) = y : (-z)$



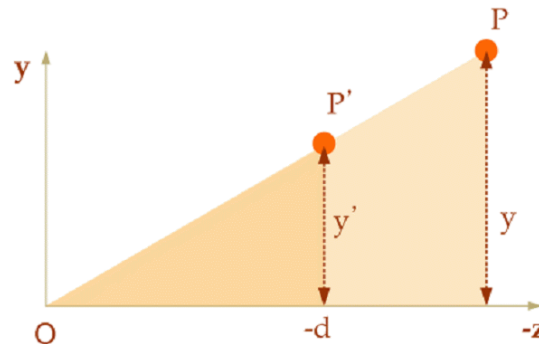
$$P' = \begin{pmatrix} x' \\ y' \\ -d \end{pmatrix} = \begin{pmatrix} \frac{x}{z/d} \\ \frac{y}{z/d} \\ -d \end{pmatrix}$$

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

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



(a)



(b)

## 3.2.1 원근 변환

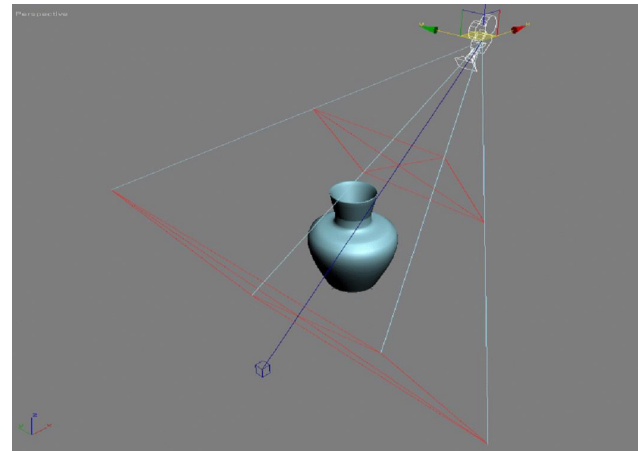
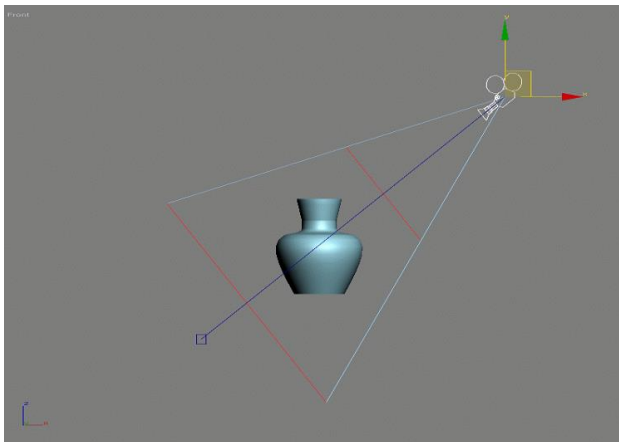
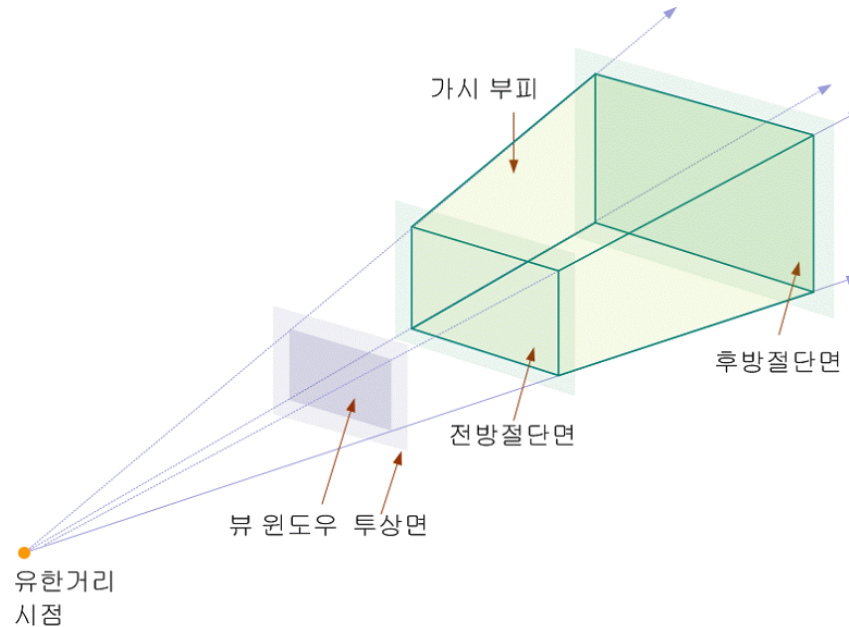
- 원근분할(Perspective Division, Homogenization)
  - 동차좌표의 마지막 요소로 이전 요소를 나누는 작업
  - 절단이 동차좌표에서 이루어지기 때문에 절단 이후로 미루어 짐.
- 원근변환
  - 어파인 변환이 아님: 마지막 행이 (0, 0, 0, 1)이 아님
  - 3차원 좌표관점:  $x' = x/(z/d)$ : 비선형 변환
  - 4차원 동차좌표 관점: 선형변환

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

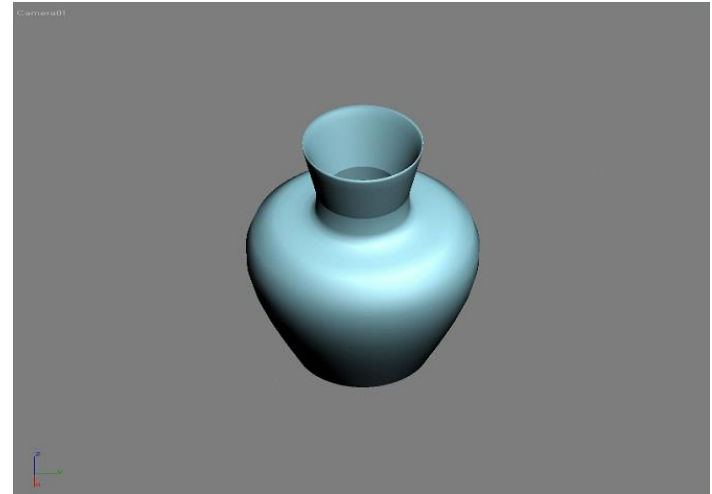
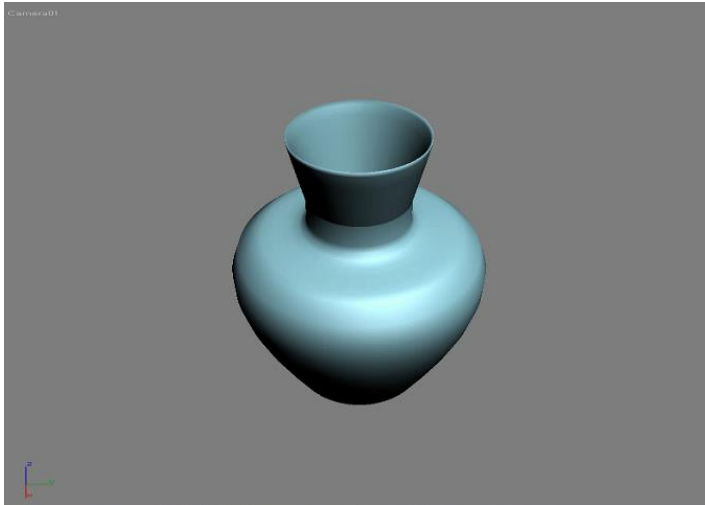
$$\begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & -1 & 0 \\ 0 & 0 & 1/d & 0 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = M_{\text{perspective}} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

### 3.2.2 정규화 가시부피 변환

- 절단 사각뿔(Frustum) = 절두체



### 3.3 원근 투상과 평행 투상



## 3.4 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

