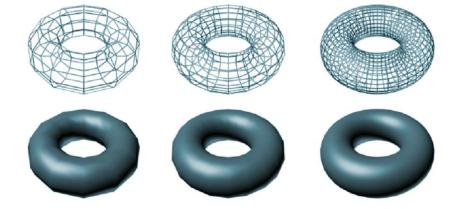
### Note 04-Transformation

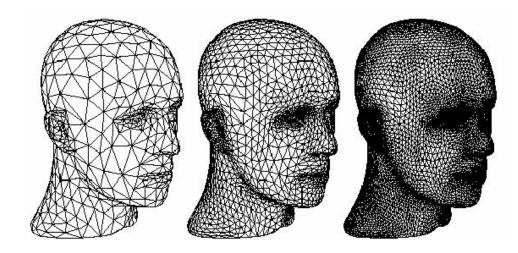
Metaverse

### 좌표계

- 3차원 물체의 표현
  - ♣ 경계면 표현(Boundary Surface Representation)
    - 메시(Mesh), 표면 메시(Surface Mesh), 다각형 메시(Polygon Mesh), 표면 다각형 (Surface Polygon), 다각형(Polygon) 등.. 다양한 명칭..
    - 사각형 메시(Rectangular Mesh): 평면 보장 못함.
    - 삼각형 메시(Triangular Mesh): 평면 보장. 2배의 드로잉 속도

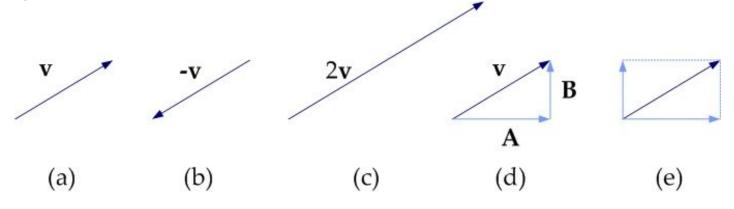


#### Level of detail (LOD)

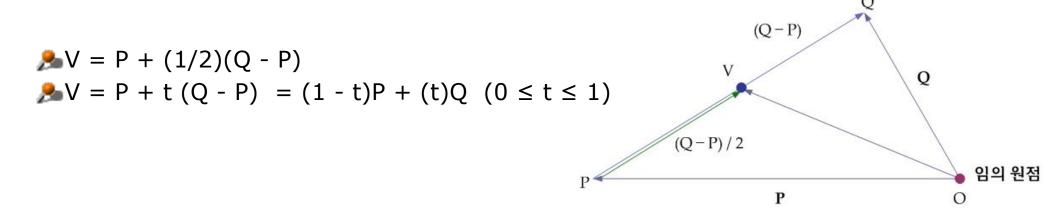


### 좌표계

• 벡터 공간



• Affine space: 점과 벡터의 동시처리..



### 좌표계

(2, 4, 2)

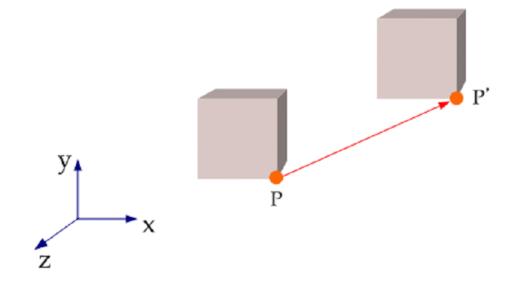
(1, 2, 1)

(1, 2)

#### • 동차좌표

- 🏂벡터와 점의 표현이 다름
  - $\bullet V = 4 V1 + 2 V2 + V3$
  - $\bullet$  P = r + 4V1 + 2V2 + V3
- 🎤 차원을 하나 올리면 동일 방법으로 표현
  - v = 4V1 + 2V2 + V3 + 0 r = (4, 2, 1, 0): 벡터
  - P = 4V1 + 2V2 + V3 + 1r = (4, 2, 1, 1): A
- №3차원 점 (1, 2, 1)
  - 4차원 동차좌표로 사상
  - 동차좌표 (1, 2, 1, 1) = (2, 4, 2, 2) = (3, 6, 3, 3) = ...
  - 동차좌표 (x, y, z, w) => 3차원 좌표 (x/w, y/w, z/w)

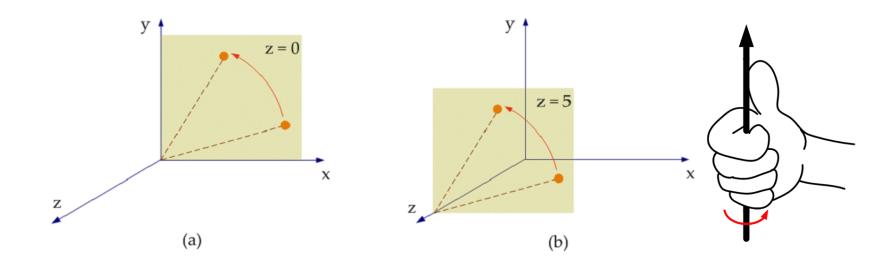
#### **Translation**



$$P' = T \cdot P$$

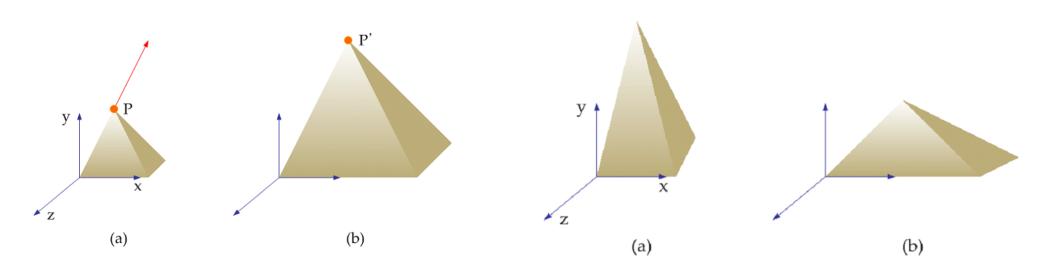
$$\begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix} = \begin{pmatrix} 1 & 0 & 0 & Tx \\ 0 & 1 & 0 & Ty \\ 0 & 0 & 1 & Tz \\ 0 & 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix}$$

#### **Rotation**



$$P' = Rz(\theta) \cdot P$$

### **Scaling**

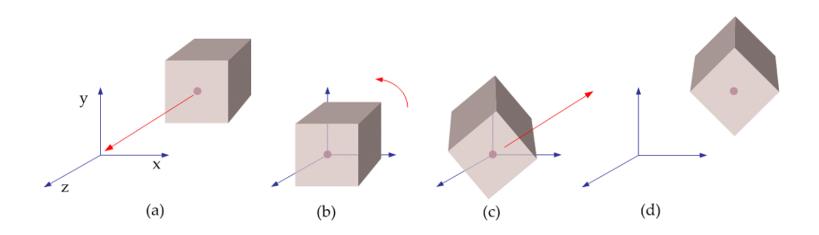


$$P' = S \cdot P$$

복합변환 (Composite Transformation)

♣크기조절(S1) 후, 결과 물체를 회전(R1)한 후, 다시 크기조절(S2)

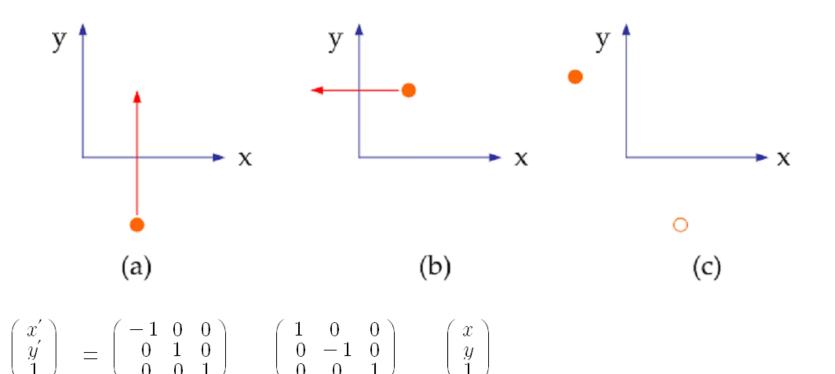
- $P' = S2 \cdot R1 \cdot S1 \cdot P$
- 행렬곱셈의 순서에 유의
- P' = C · P 복합행렬 C는 한번만 계산. 모든 정점에 적용



$$C = T(Xp, Yp, Zp) \cdot Rz(\theta) \cdot T(-Xp, -Yp, -Zp)$$

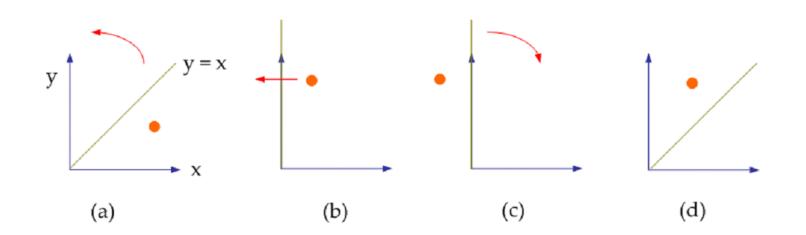
8개의 vertex: vi vi'=Cvi

• 반사 (Reflection)



$$C = \begin{pmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad = \begin{pmatrix} -1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

• 복합변환 반사 (Reflection)



$$\begin{pmatrix} \cos(-45°) - \sin(-45°) & 0\\ \sin(-45°) & \cos(-45°) & 0\\ 0 & 0 & 1 \end{pmatrix}$$

$$\begin{pmatrix} \cos{(-45°)} - \sin{(-45°)} & 0 \\ \sin{(-45°)} & \cos{(-45°)} & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \qquad \begin{pmatrix} \cos{(45°)} - \sin{(45°)} & 0 \\ \sin{(45°)} & \cos{(45°)} & 0 \\ 0 & 0 & 1 \end{pmatrix}$$

#### 3차원은?

- 변환의 분류
- ♣강체변환(Rigid Body Transformation)
  - 이동변환, 회전변환
  - 물체 자체의 모습은 불변
- 🔑 유사변환(Similarity Transformation)
  - 강체변환 + 균등 크기조절 변환, 반사변환
  - 물체면 사이의 각이 유지됨.
  - 물체내부 정점간의 거리가 일정한 비율로 유지됨

#### ♪ 어파인변환(Affine Transformation)

2차원 공간에서 2x3 matrix를 사용한 이미지 변환

- 유사변환 + **차등 크기조절 변환, 전단(shearing) 변환**
- 물체의 타입이 유지
  - 직선은 직선으로, 다각형은 다각형으로, 곡면은 곡면으로
  - 평행선이 보존
  - 변환행렬의 마지막 행이 항상 (0, 0, 0, 1)

(	1	0	0	Tx	1
	0	1	0	Ty	
	0	0	1	Tz	
	0	0	0	1	J

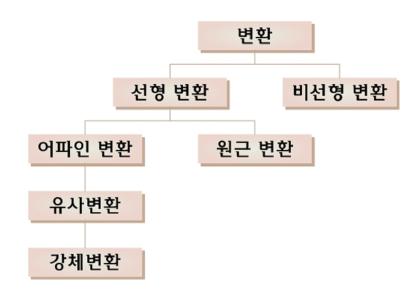
#### • 변환의 분류

- ♣원근변환(Perspective Transformation)
  - 평행선이 만남.
  - 직선이 직선으로 유지
  - 변환행렬의 마지막 행이 (0, 0, 0, 1) 아님.

### ♣️선형변환(Linear Transformation)

- 어파인 변환 + 원근 변환
- 선형 조합(Linear Combination)으로 표시되는 변환
- x' = ax + by + cz에서 x'는 x, y, z 라는 변수를 각각 상수 배 한 것을 더한 것이다.
- 예:

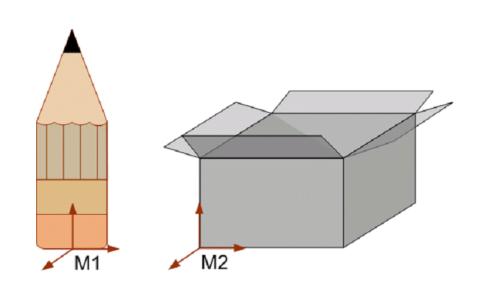
$$\begin{aligned} x^{'} &= r cos \left(\phi + \theta \right) = r cos \phi cos \theta - r sin \phi sin \theta = x cos \theta - y sin \theta \\ x^{'} &= x + Shy \cdot y \end{aligned}$$



### 모델 변환

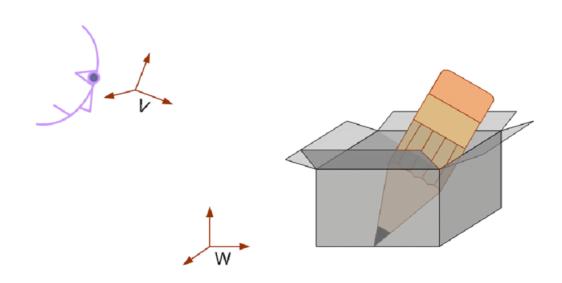
- Modeling: 물체의 형태를 설계 --> 물체를 구성하는 Vertex의 위치를 정의 --> 원점, 좌표축 --> 좌표계!
  - 좌표계의 원점 및 축: 물체마다 설계상의 편의를 위해 다르게 구성
- Model Coordinate System (MCS)
  - or Local Coordinate System

MCS: 단일 모델의 메쉬를 정의하기 위해서 만들어진 좌표계!



### 모델 변환

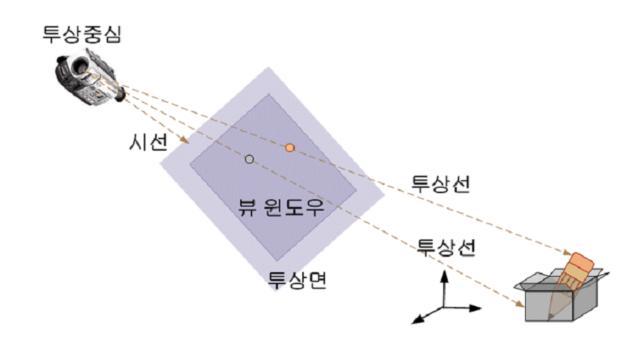
- World Coordinate System (WCS)
  - 여러물체가 존재 = 여러 지역 좌표계가 존재
  - 일률적으로 어우를 수 있는 기준 좌표계 WCS
  - 임의의 위치에 선정



• 시점 좌표계(VCS, View Coordinatge System)

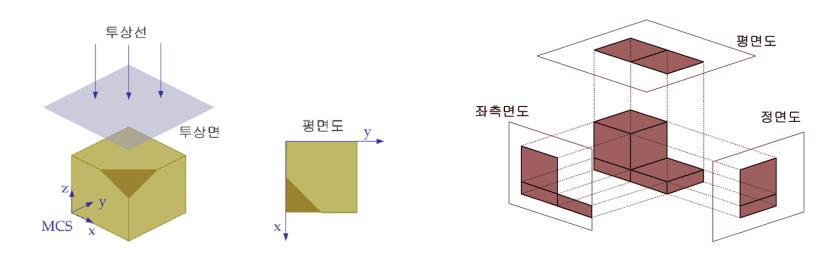
## Projection

Projection = 가시변환 (Viewing Transformation)

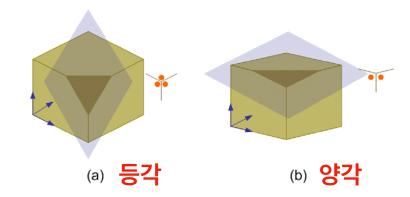


- 투상면(View Plane, Projection Plane) : 필름과 유사
- 관찰자 위치(View Point, Eye Position) = 카메라 위치(Camera Position)
  = 투상중심(COP: Center of Projection) = 시점좌표계 원점(Origin of VCS)
- 투상선(Projectors): 물체 곳곳을 향함
- 시선(Line of Sight): WCS원점 또는 초점을 향함 (카메라의 look at 방향)

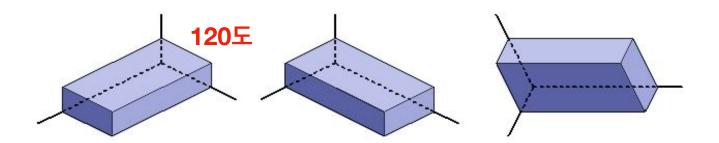
- Orthographic Projection
- ♪ 평면도, 입면도, 측면도 등
  - 주 평면(Principal Plane): MCS (or WCS) 주축인 x, y, z에 의해 형성되는 x-y, y-z, z-x
  - 투상면은 주 평면 중 하나와 평행
- 🥦 투상선은 투상면과 직교
  - 원래 물체의 길이를 정확히 보존. 공학도면에 사용
  - 투상선이 반드시 투상면과 직교-> 시점위치가 제한됨.



- Axonometric Projection (축측투상)
  - 🏂 한꺼번에 여러 면을 보여줌
    - 투상선은 투상면(view plane) 과 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 대 칭적으로 놓일 때. 3개의 축 방향에 대해 동일 축소율
      - 정삼각형-> 정삼각형으로 투영



- Axonometric Projection (축측투상)
  - 등각(동형, Isometric)
    - 투상면이 3 개의 주 평면이 만나는 모서리에서 모든 평면에 대해 대 칭적으로 놓일 때. 3개의 축 방향에 대해 동일 축소율
    - 정삼각형-> 정삼각형으로 투영

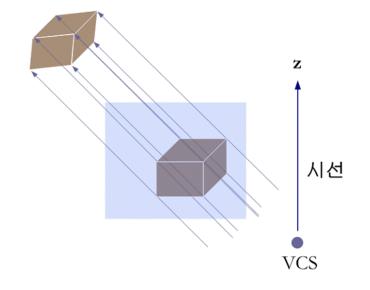


• Oblique Projection (경사투상)

🥕투상선끼리는 평행

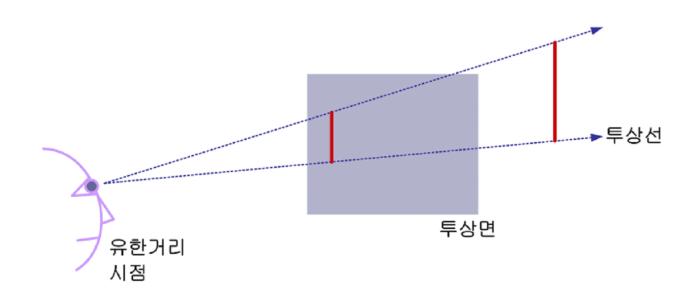
♣투상면은 시선에 수직이지만 투상선과 직교하지 않음.

♣고개는 돌리지 않고 눈동자만 돌려서 보는 것과도 흡사



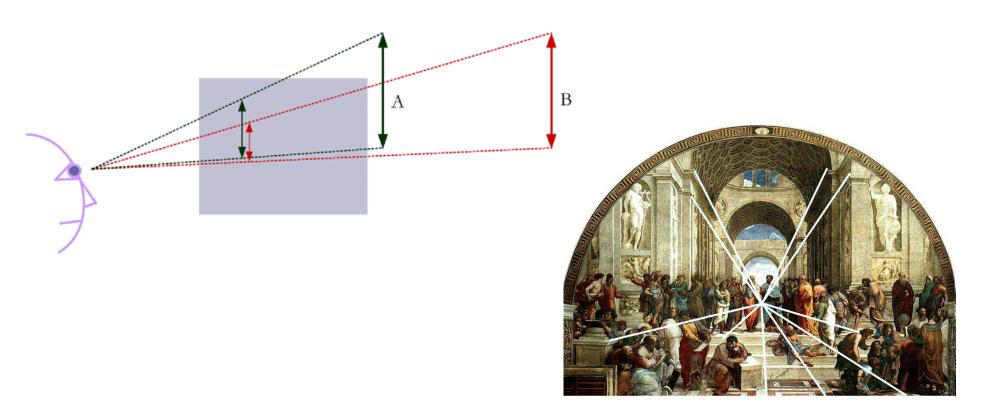
• 그래픽스에서 (특히 Computer Aided Design) 주로 사용하는 방식은..Orthographic Projection

- ♣시점이 물체로부터 유한한 거리에 있다고 간주
- ♣투상선이 시점에서 출발하여 방사선 모양으로 퍼져감.
- ♪ 카메라나 사람의 눈이 물체를 포착하는 방법 (일반적인 어플리케이션 (게임, VFX등) 사용)



#### ♪ 원근감(Depth Feeling)

● 동일한 크기의 물체라도 시점으로부터 멀리 있는 것은 작게 보이고 가까운 것은 크게 보임



선형원근 (Linear Perspective --> 깊이감 인지)

Sub note: Aerial Perspective



By Leonardo da Vinci

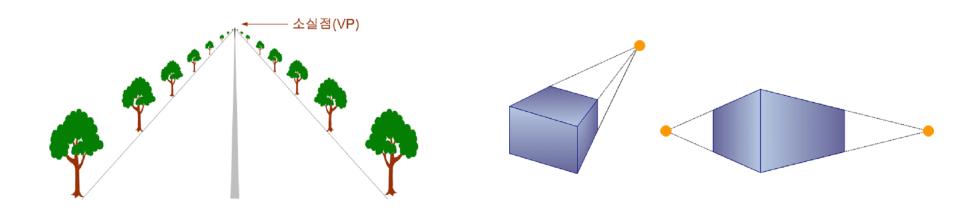
colors become less distinct over distance



in computer graphics.. environmental fog

Fig. 1.3 Shadow in the Summer (© 2005 Stephan Martiniere)

- ♣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)



### Viewport Transformation

- ♣ 정규화 장치좌표계(NDCS: Normalized Device Coordinate System)
  - 절단 이후 원근분할에 의해 물체 정점을 3차원 좌표로 변환한 것
  - $\bullet$  (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)

