

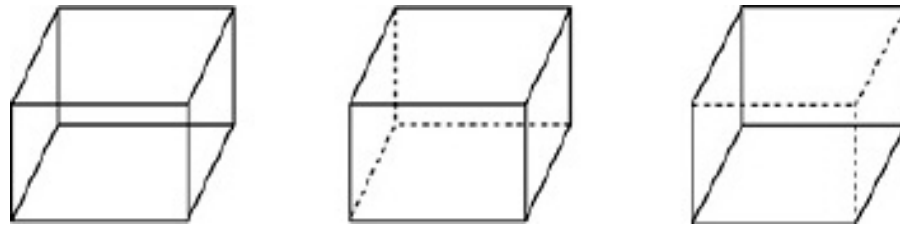
은면 제거

동아대학교 컴퓨터공학과
박영진



은면제거의 개념

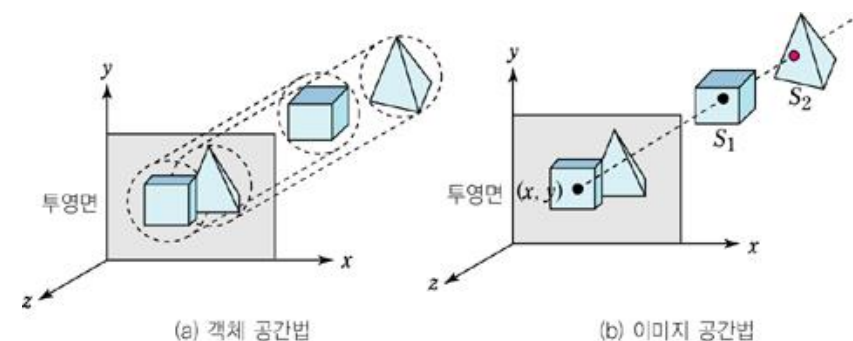
- 일반적인 3D 공간에는 다양하고 많은 객체가 존재할 수 있음
 - 이를 2D 화면에 투영하여 디스플레이
- 다른 객체나 면들에 가려져 보이지 않는 면들을 제거 : 은면 제거
- 관측자의 시점에 따라 다른 결과
- 은면제거의 목적 : 실감 있는 그림의 생성
 - 은선제거(Hidden Line Removal), 은면제거(Hidden Surface Removal)



육면체의 은선제거 효과

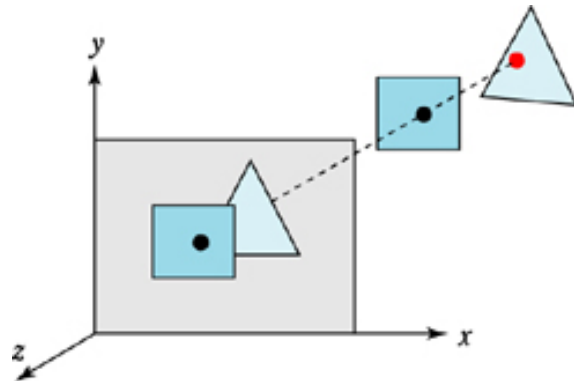
은면제거 알고리즘 기법의 구분

- 객체공간법(Object space method)
 - 공간상 객체의 위치관계를 이용하여 은면 결정
 - 객체의 수가 적거나, 객체들이 서로 분산되어 존재하는 경우 효율적인 방법
 - 수가 많은 경우 많은 처리 시간이 소요
 - 깊이 정렬 알고리즘 등
- 이미지공간법(Image space method)
 - 투영된 픽셀평면에서 객체가 보이는지 여부를 검사
 - 각 픽셀로부터 객체들로 향한 투영선을 그었을 때, 처음으로 관통(Hit)하는 객체의 면이 보인다는 사실 활용
 - z-버퍼 알고리즘 등

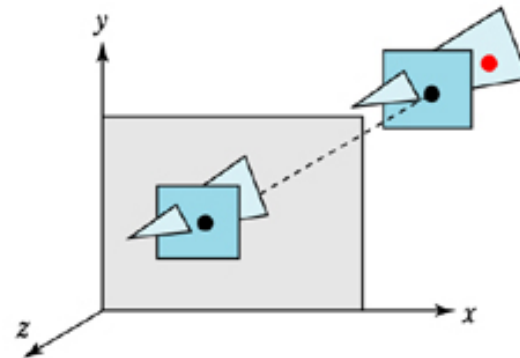


은면제거의 처리에 적용되는 일반적인 원리

- 객체의 각 표면을 거리에 따라 정렬(Sorting)
 - 투영면으로부터 가장 가까이 위치한 객체 순으로 정렬하거나, 수평, 수직방향으로 객체들을 정렬
 - 투영면에 가장 가까운 면을 선택하여 그린다.
 - 대표적인 예: 깊이정렬법



(a) 객체가 정렬되어 있는 경우

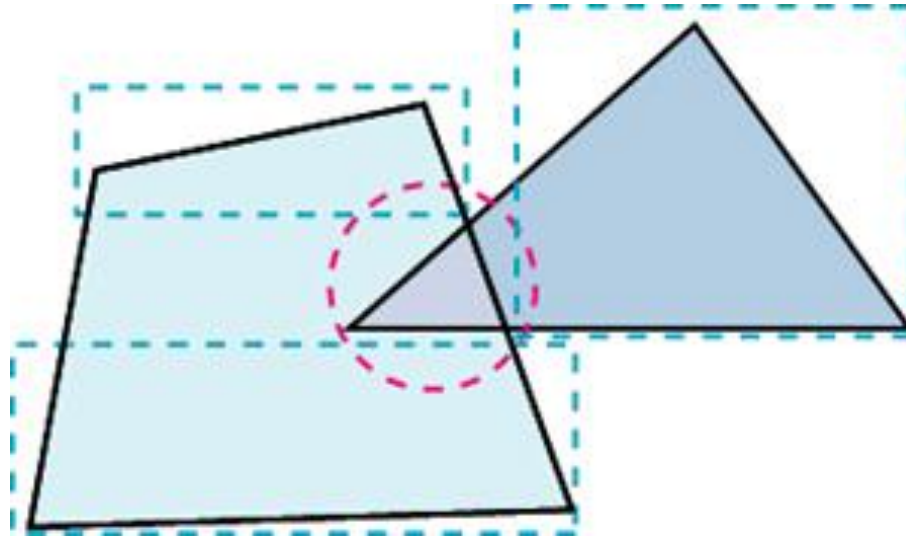


(b) 별도의 처리과정이 필요한 경우

객체의 정렬을 이용한 은면 제거

은면제거의 처리 개념

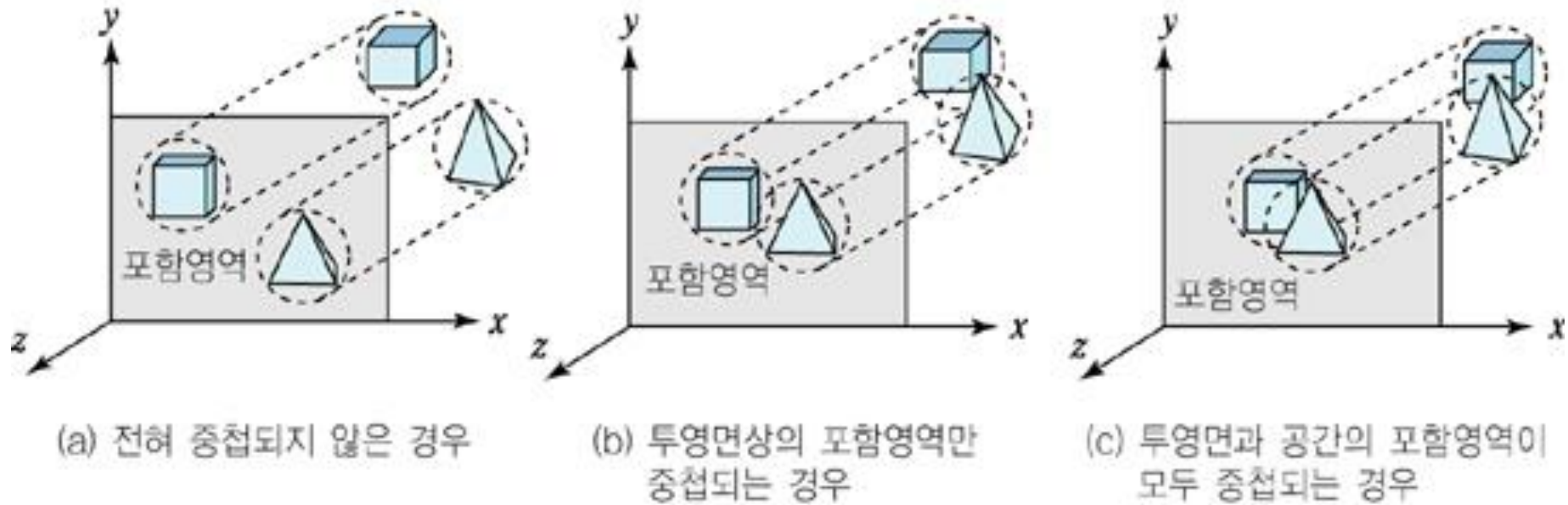
- 객체의 일관성(coherence) 이용
 - 공간적 일관성 이용하여 효율 증대
 - 애니메이션에서는 시간적 일관성 이용



공간적 응집성의 예

은면제거의 처리 개념

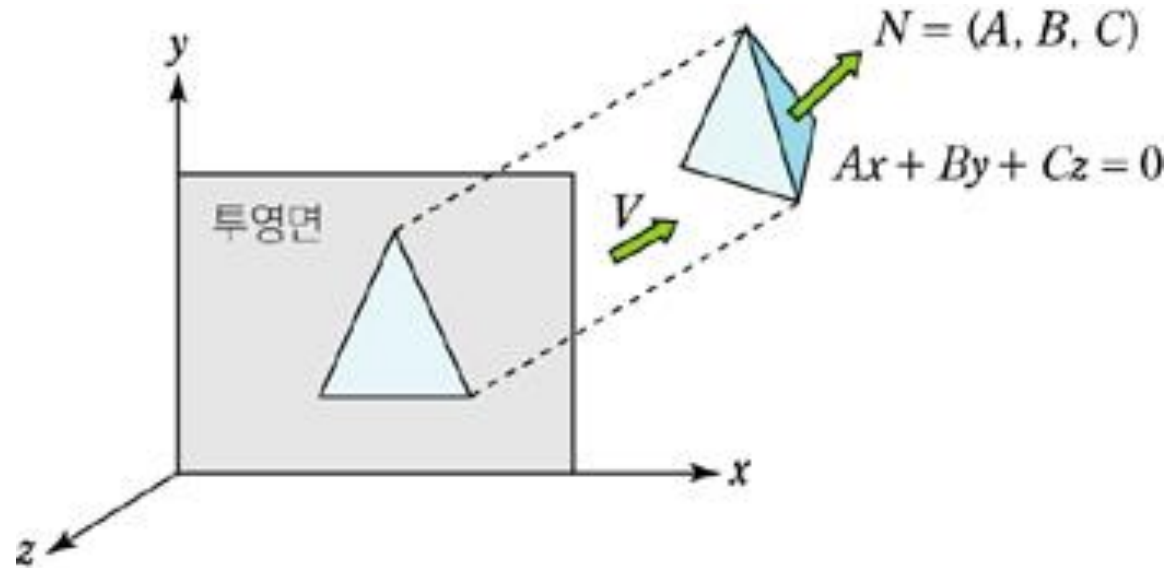
- 포함영역(Extent, Bounding Volume)를 이용하여 비교횟수 축소
 - 우선, 간단한 모양의 포함영역 간의 겹침을 비교



공간객체와 포함영역(Extent)과의 관계

다면체 뒷면의 제거(Back-Face Removal)

- 은면제거의 첫 단계에서 이용
 - 빠르고 간단하게 뒷면을 찾는 객체공간 방법
 - 일반적인 장면에서 다면체 면의 약 50% 제거

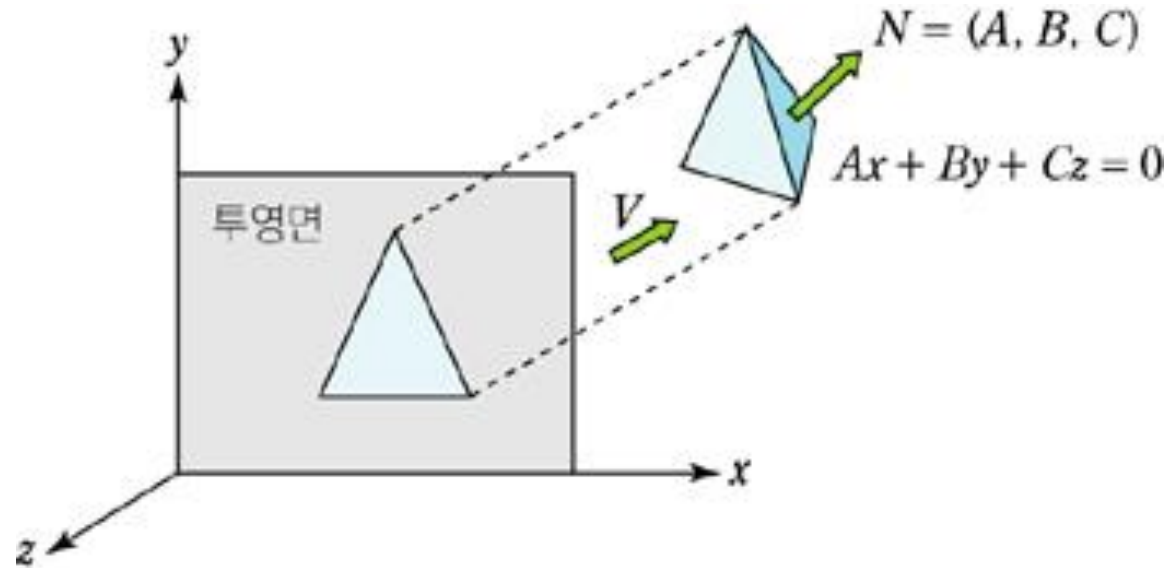


다면체 뒷면의 결정

다면체 뒷면의 제거(Back-Face Removal)

- 평면방정식 이용

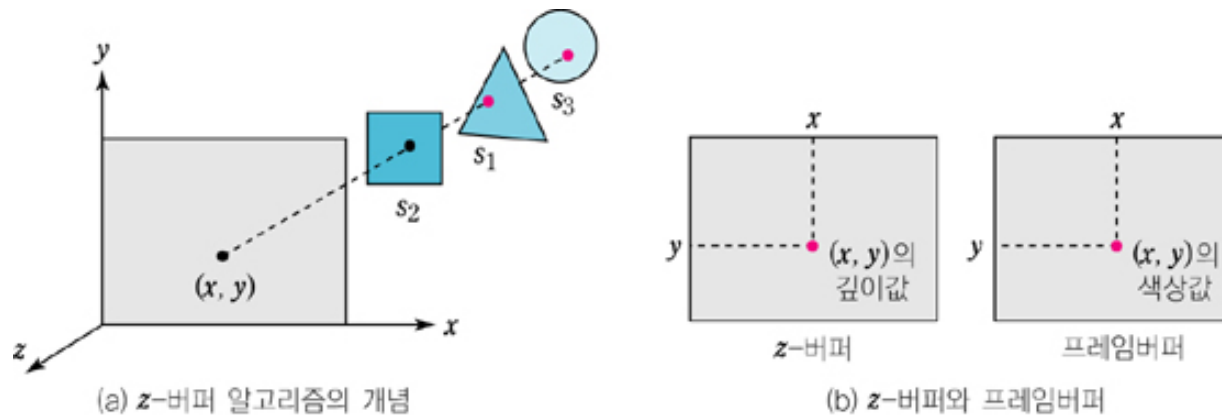
- $Ax + By + Cz + D = 0$ 에서 법선벡터는 $N = (A, B, C)$
- 시선방향 벡터 V 와 $V \cdot N > 0$ 이면 뒷면, $V \cdot N < 0$ 이면 앞면



다면체 뒷면의 결정

z-버퍼(z-Buffer) 기법

- z-버퍼 기법의 개념
 - 가장 일반적으로 사용되는 이미지 공간 접근방법
 - 깊이버퍼(Depth Buffer) 알고리즘이라고도 부른다.
- 물체의 가시성을 픽셀 단위로 조사
 - z 값이 가장 작은 평면의 값을 그린다
- z 값을 저장하는 깊이버퍼 (z-버퍼) 와 색상을 저장하는 프레임 버퍼가 필요



z-버퍼 알고리즘

z-버퍼(z-Buffer) 기법

- 특징

- 일반 곡면에 대해서도 적용 가능
- 구현이 간단하며 다각형 면에 대한 정렬이 필요 없음
- 추가버퍼가 요구되므로 이를 위한 저장 공간이 필요

		0.6			
	0.7	0.7			
0.8	0.8	0.8	0.8		
0.9	0.9	0.9	0.9	0.9	

이미지1

0.9	0.9	0.9	0.9	0.9	
	0.9	0.9	0.9		
		0.9			

이미지2

		0.6			
	0.7	0.7			
0.8	0.8	0.8	0.8		
0.9	0.9	0.9	0.9	0.9	

이미지1

+

1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0

깊이버퍼

=

1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	0.6	1.0	1.0	1.0
1.0	0.7	0.7	1.0	1.0	1.0
0.8	0.8	0.8	0.8	1.0	1.0
0.9	0.9	0.9	0.9	0.9	1.0

결과

0.9	0.9	0.9	0.9	0.9	
	0.9	0.9	0.9		
		0.9			

이미지2

+

1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	1.0	1.0	1.0	1.0
1.0	1.0	0.6	1.0	1.0	1.0
1.0	0.7	0.7	1.0	1.0	1.0
0.8	0.8	0.8	0.8	1.0	1.0
0.9	0.9	0.9	0.9	0.9	1.0

깊이버퍼

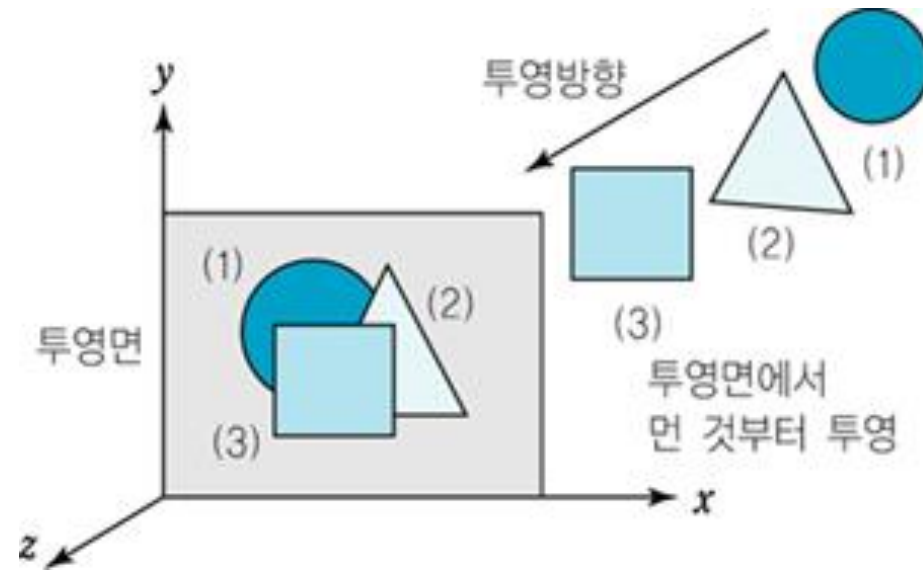
=

1.0	1.0	1.0	1.0	1.0	1.0
0.9	0.9	0.9	0.9	0.9	1.0
1.0	0.9	0.6	0.9	1.0	1.0
1.0	0.7	0.7	1.0	1.0	1.0
0.8	0.8	0.8	0.8	1.0	1.0
0.9	0.9	0.9	0.9	0.9	1.0

결과

깊이정렬법

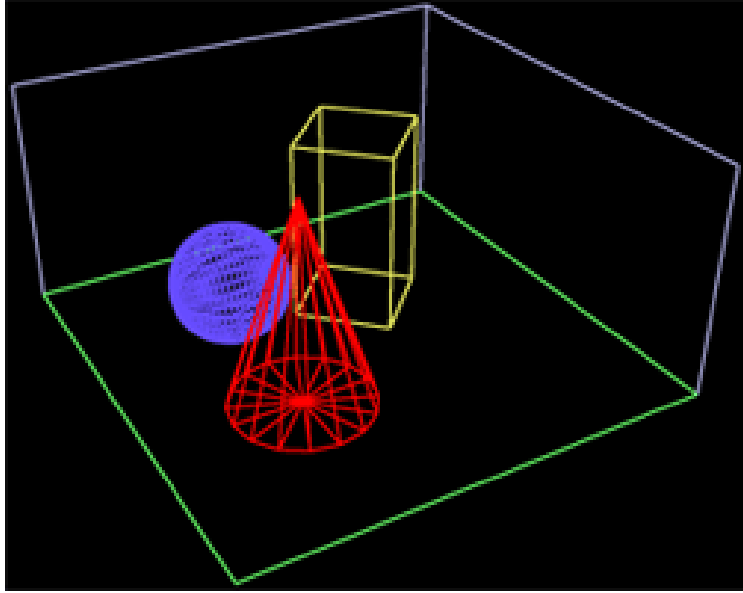
- 페인터 알고리즘(Painter's Algorithm)
 - 다각형면을 깊이(z값)에 따라 정렬한 뒤, 먼 것부터 투영하여 그린다.
 - 뒤쪽에 있는 면은 나중에 그린 면(가까운 면)에 가려진다



깊이 정렬 기법

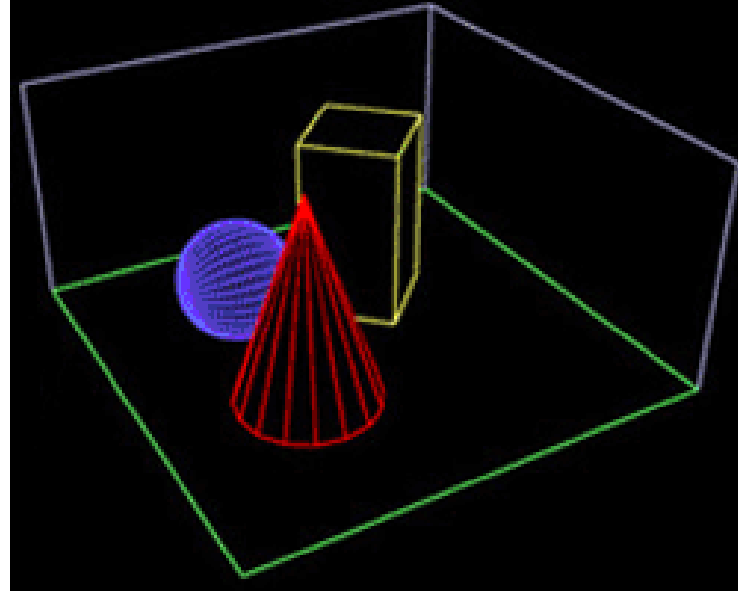
곡면 렌더링(Surface Rendering)의 과정

- 와이어프레임 모델링



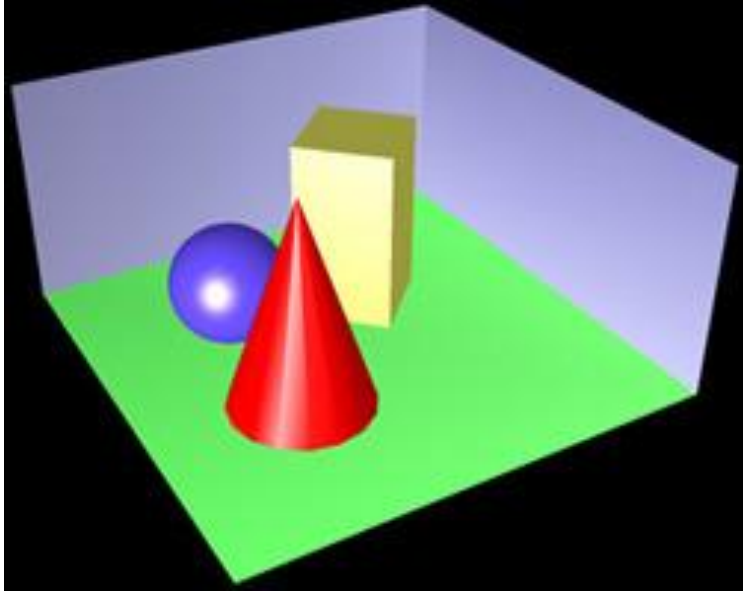
→

은면 제거

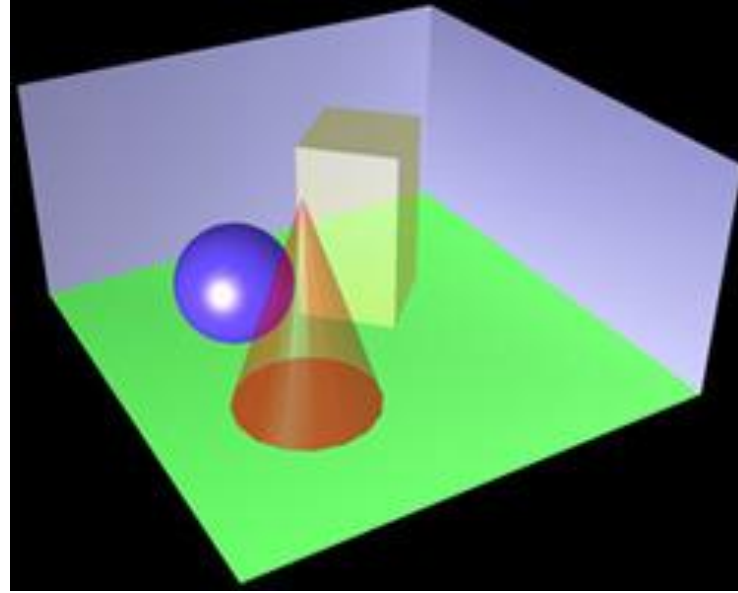


곡면 렌더링(Surface Rendering)의 과정

- 면의 셰이딩



→ 투명한 물체의 표현



곡면 렌더링(Surface Rendering)의 과정

- 텍스처 매핑

→ 그림자 생성

