

Part2: Rendering

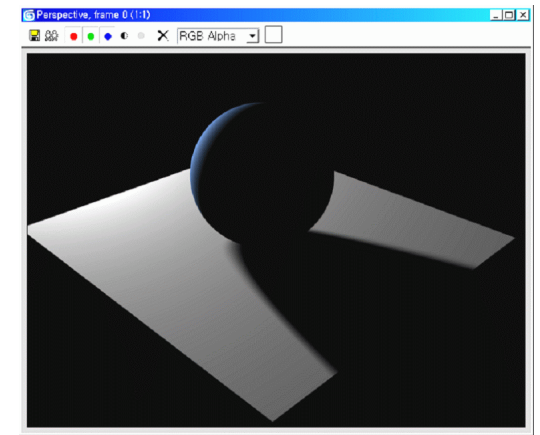
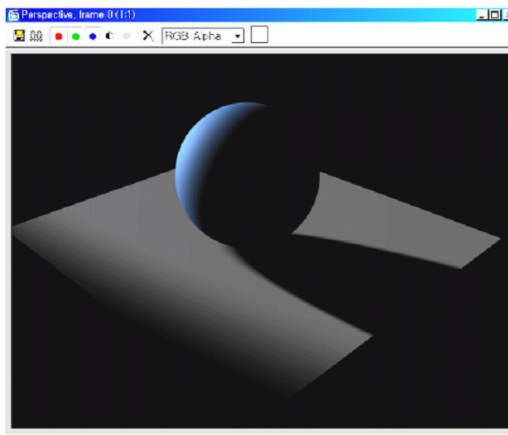
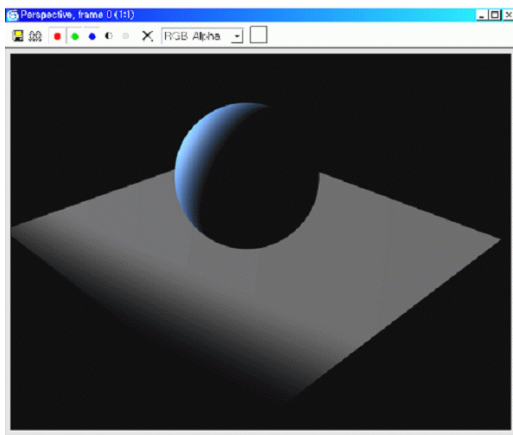
6. shadow

Outline

- I. 그림자
- II. 지면 그림자
- III. 쉼도우 맵
- IV. 그림자 부피

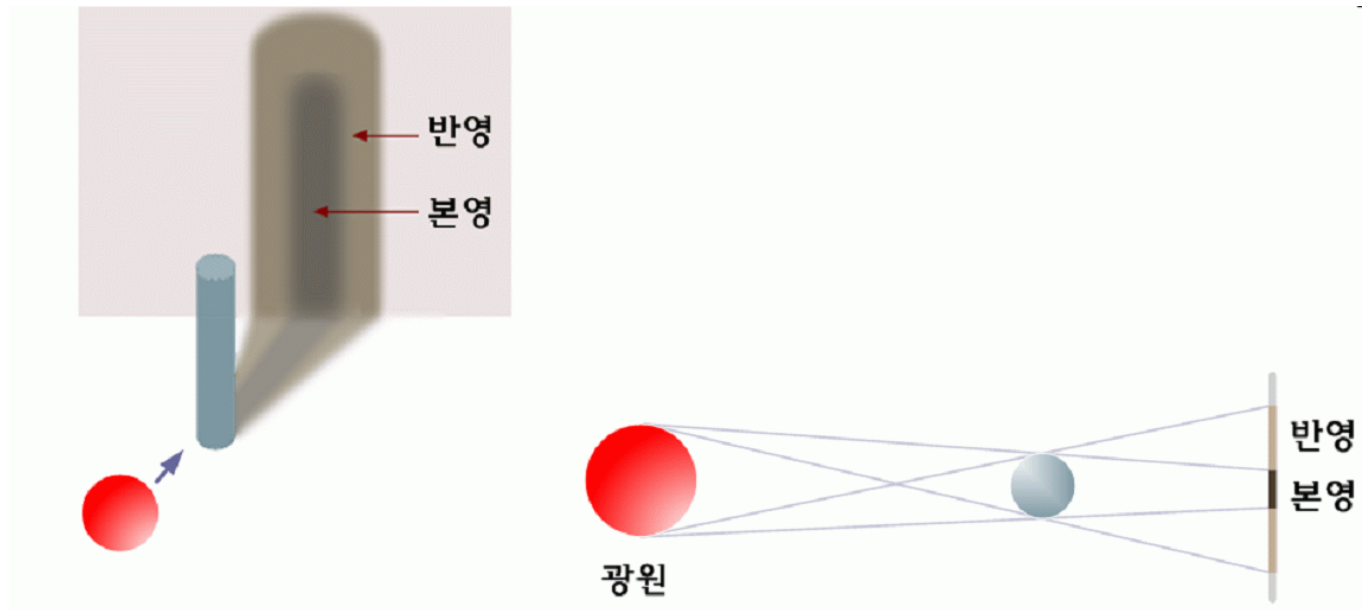
1. 그림자

- 셰이드(Shade)
- 그림자
 - 셰도우(Shadow)와 구별
 - 상대적으로 단순한 연산으로 매우 사실적인 효과
 - 조명과 별도로 취급
 - 그림자 없음: 공중에 떠있는 인상



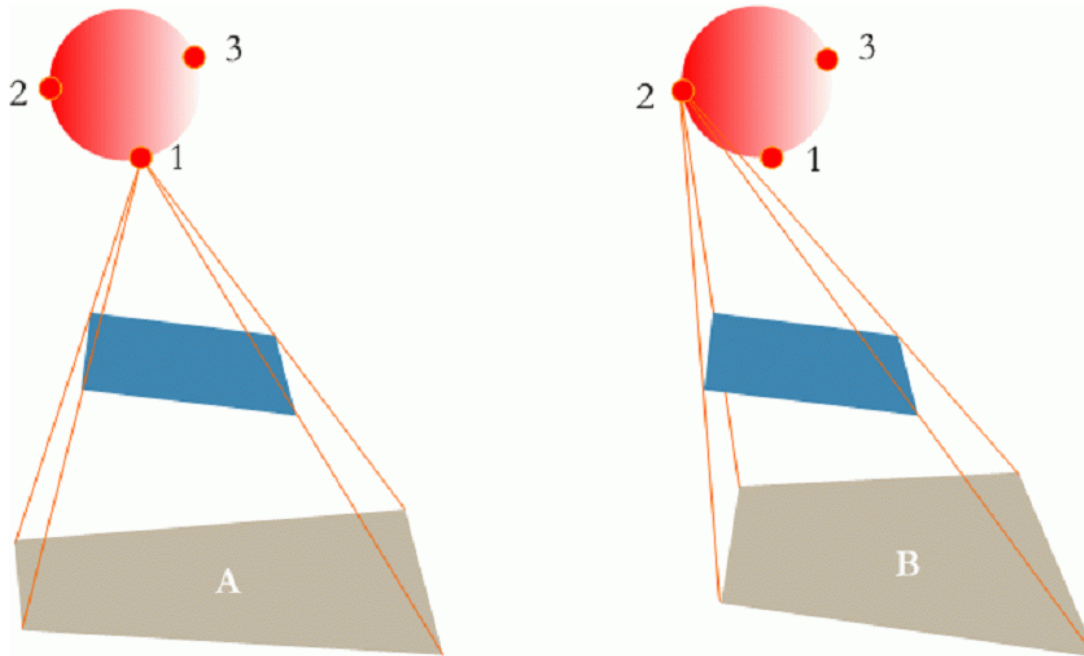
1.1 반영과 본영

- 본영(本影, 본 그림자, Umbra, Hard Shadow)
- 반영(半影, 반 그림자, Penumbra, Soft Shadow)
- 그림자
 - 주변광에 의해 보임
 - 산란으로 인해 본영과 반영의 경계선이 흐려져 보임



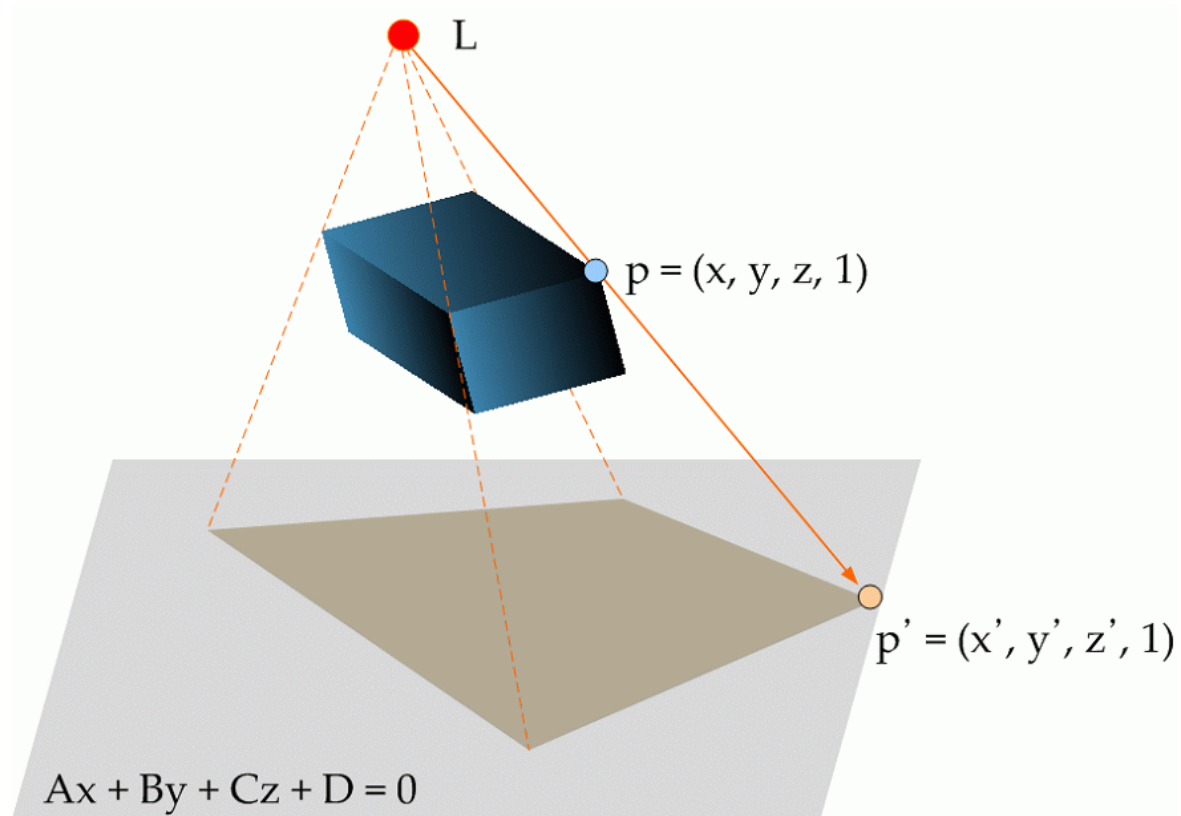
1.2 소프트 셰도우 기법

- 면적 광원의 근사화
 - 교집합: 본영
 - 합집합 - 교집합: 반영(Soft Shadow)



2. 지면 그림자

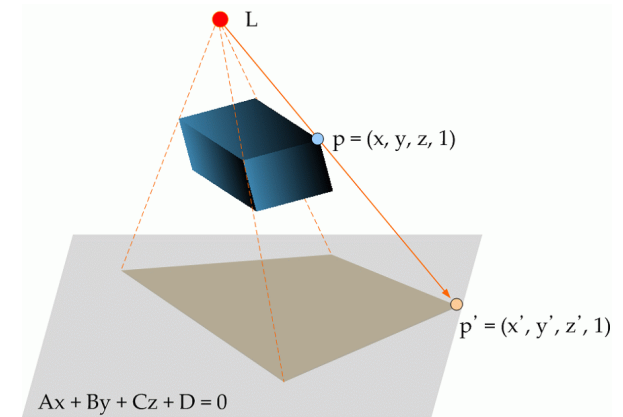
- 지면 그림자(Ground Plane Shadows)
 - 투상 그림자(Projection Shadows)
 - 바닥에 투상된 그림자만을 고려. 물체면 간의 그림자는 고려치 않음
 - 물체공간 알고리즘



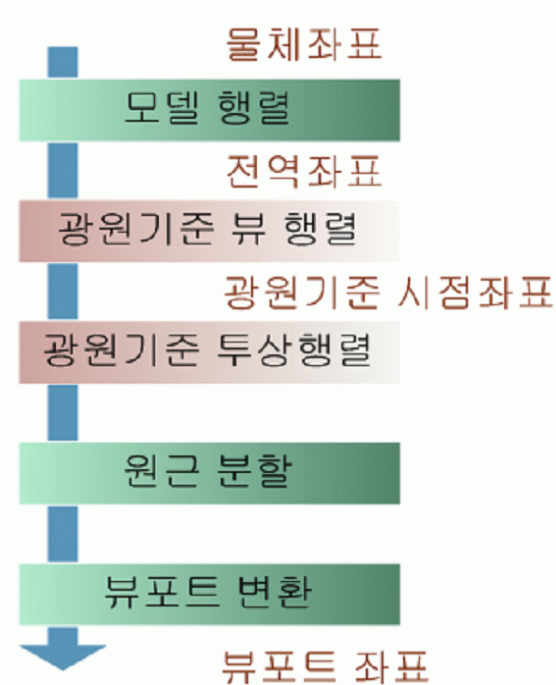
2. 지면 그림자

$$P' = T \cdot P$$

$$= \begin{pmatrix} -D & 0 & 0 & 0 \\ 0 & -D & 0 & 0 \\ 0 & 0 & -D & 0 \\ A & B & C & 0 \end{pmatrix} \cdot \begin{pmatrix} x \\ y \\ z \\ 1 \end{pmatrix} = \begin{pmatrix} x' \\ y' \\ z' \\ 1 \end{pmatrix}$$

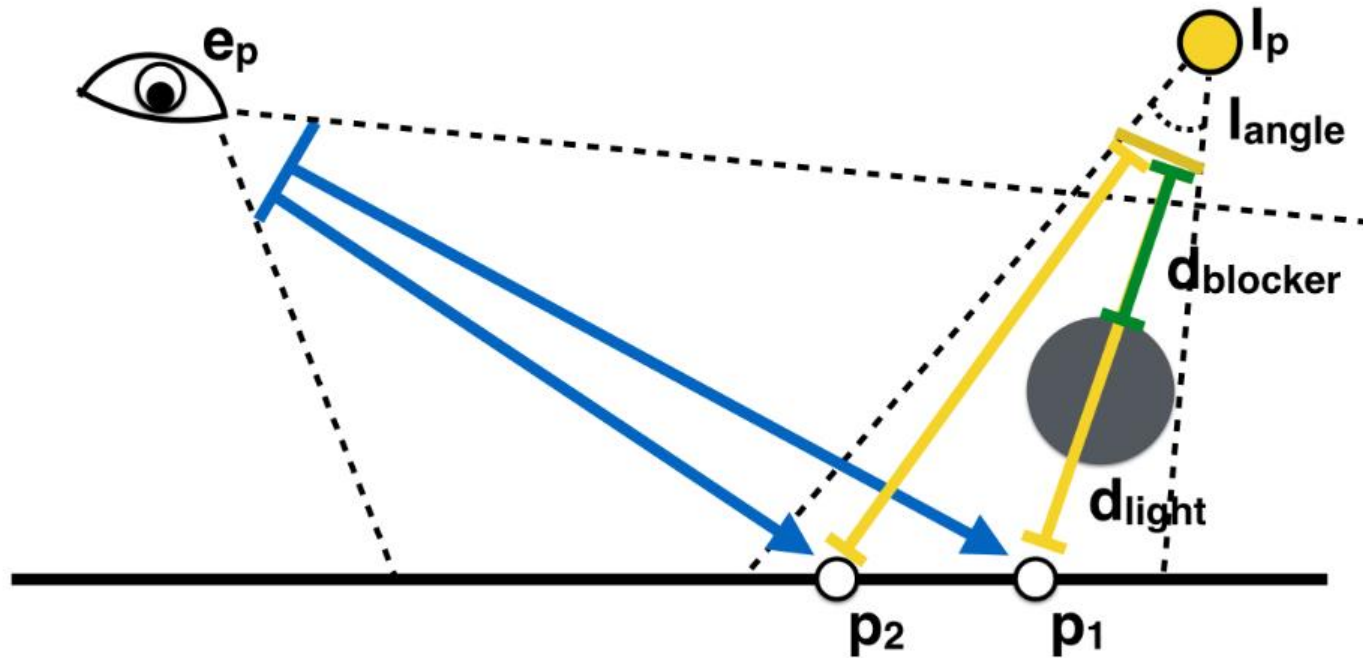


(a)



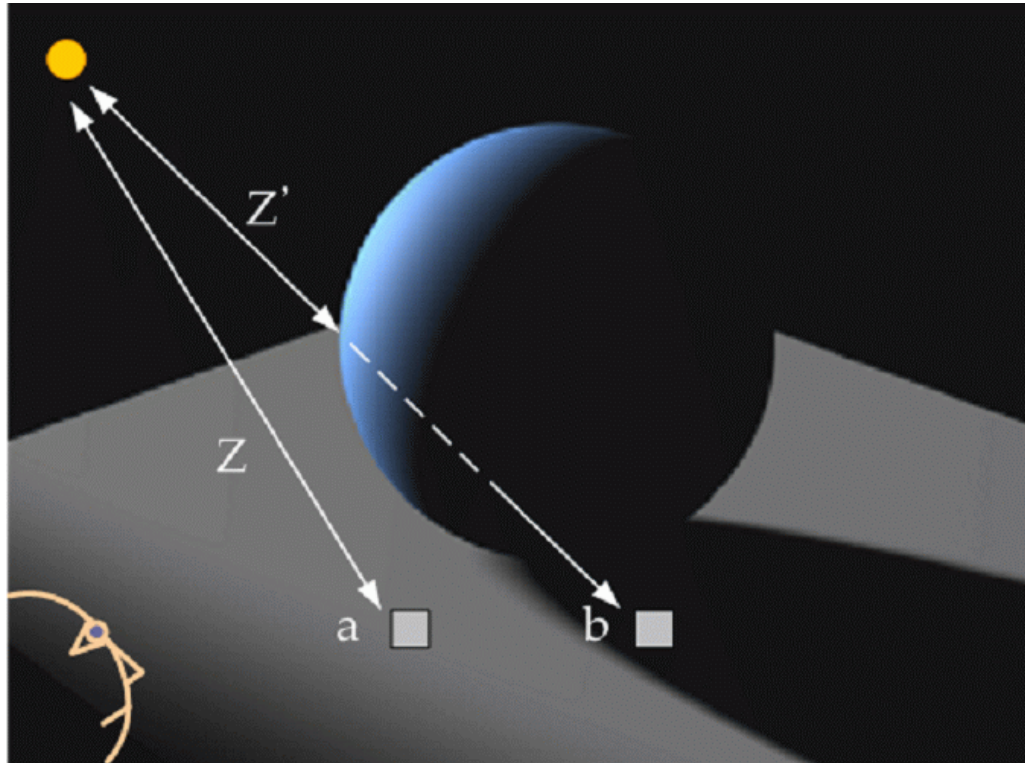
(b)

3. 셰도우 맵



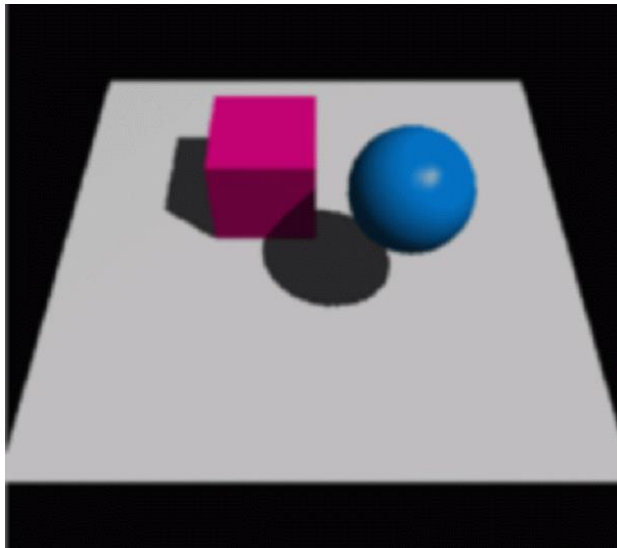
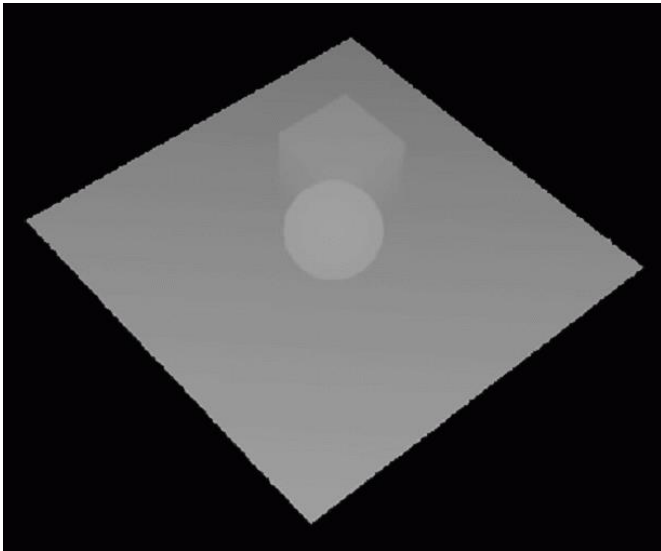
3. 셰도우 맵

- 화소 단위의 영상공간 알고리즘
 - 1. 광원기준 가장 가까운 물체면과의 거리(A)를 셰도우 지-버퍼에 저장
 - 2. 은면제거를 위해 시점기준 가장 가까운 물체면 화소를 구함
 - 3. 해당 화소 물체면으로부터 광원까지의 거리 (B)를 구함.
 - 4. B가 A보다 크면 해당 화소는 그림자 영역 내부에 존재함.



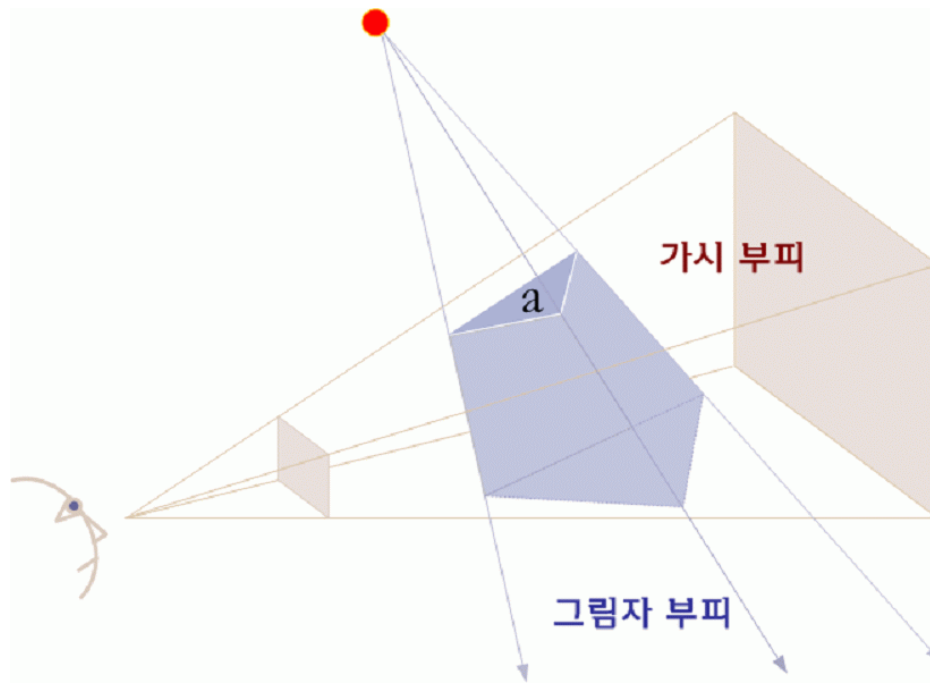
3. Shadow Map (셰도우 맵)

- if $(Z_{map} + Bias) < Z_{pixel}$
the Pixel Is in the Shadow;
else the Pixel Is Not in the Shadow;
Bias: Z_{map} 과 Z_{pixel} 이 정확히 일치하지 않을 때에 대비
- 광원기준 셰도우 맵, 시점 기준 물체 렌더링



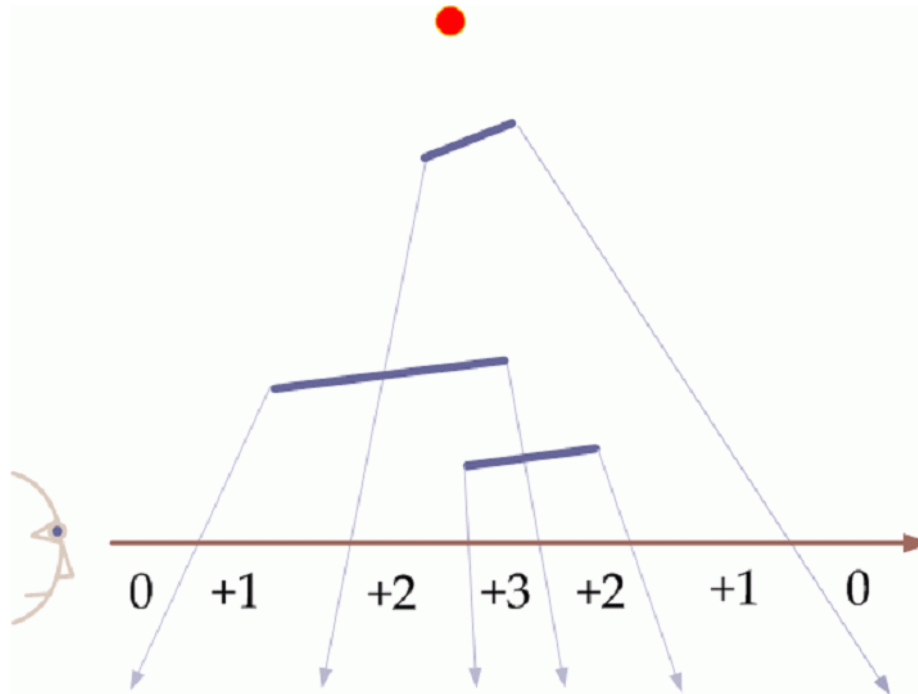
4. 그림자 부피

- 가시부피와 그림자 부피가 중첩되는 곳이 그림자 영향권



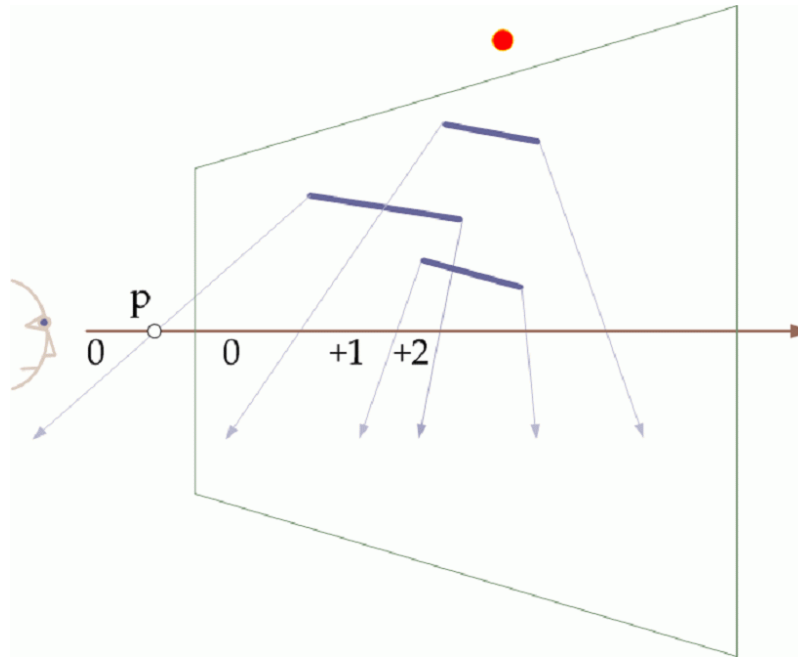
4.1 Z-pass Algorithm

- 다수개의 다각형에 의한 그림자
- 그림자 부피 안으로 들어갈 때 +1, 나갈 때 -1
- 결과적인 카운트 값이 0보다 크면 그림자 색



4.1 Z-pass Algorithm

- 전방절단면 앞쪽이 이미 그림자 부피이면 지-패스는 오류



4.2 Z-Fail Algorithm

- 지-페일 알고리즘
 - 무한대 거리에서 시점을 향해 오면서 카운트 값을 계산
 - 그림자가 후방절단면 뒤쪽에 걸칠 때엔 이 역시 오류

