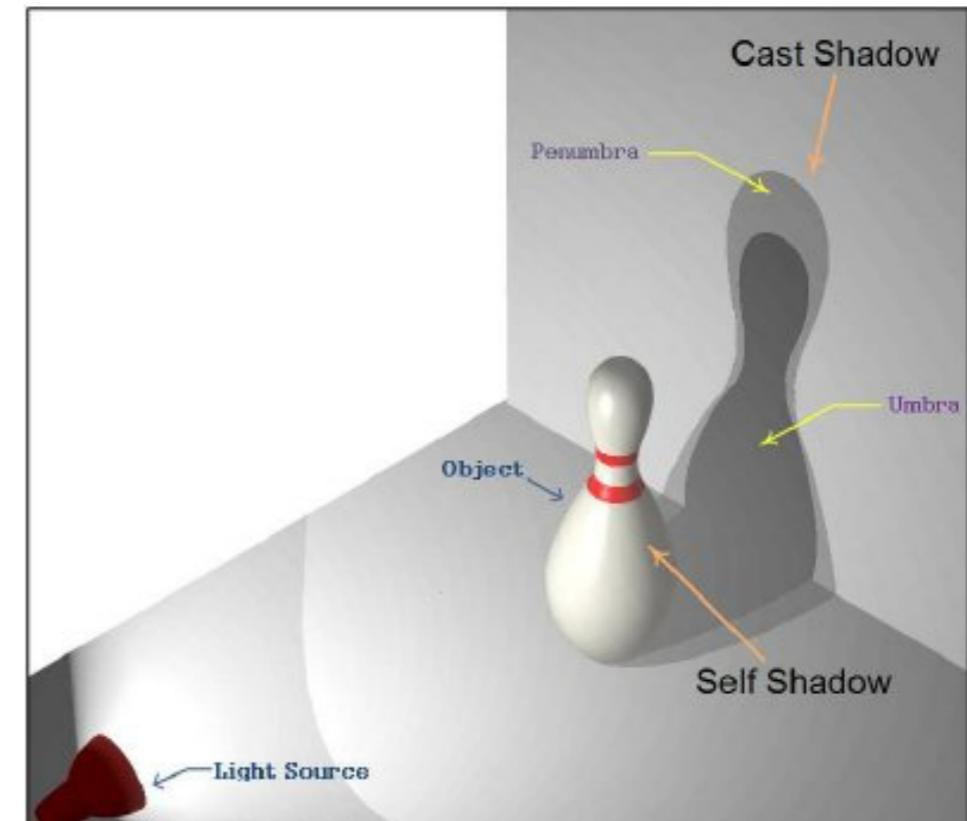


Gràfics i Visualització de Dades

Anna Puig

Ombres i penombres



Índex

2.1. Introducció: algorisme principal

2.2. Càlcul de Raig Primari

2.3. Càlcul del color:

 2.3.1. Interseccions amb objectes

 2.3.2. Materials i Llums

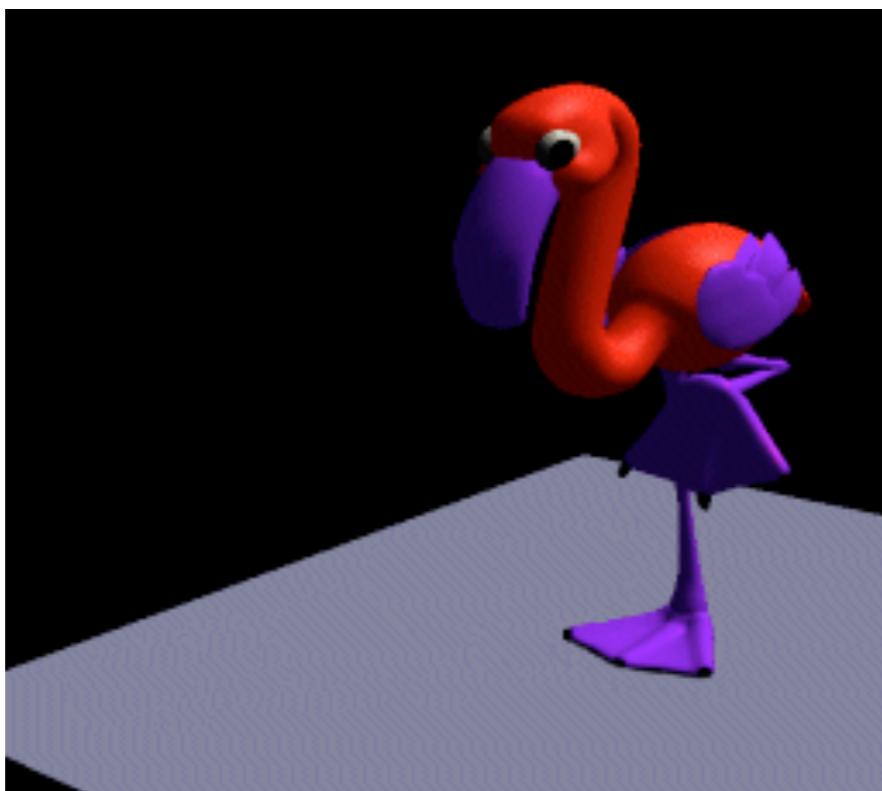
2.4. **Ombres**

2.5. Reflexions i transparències

2.6. RayTracing en models de volum

2.4. Ombres

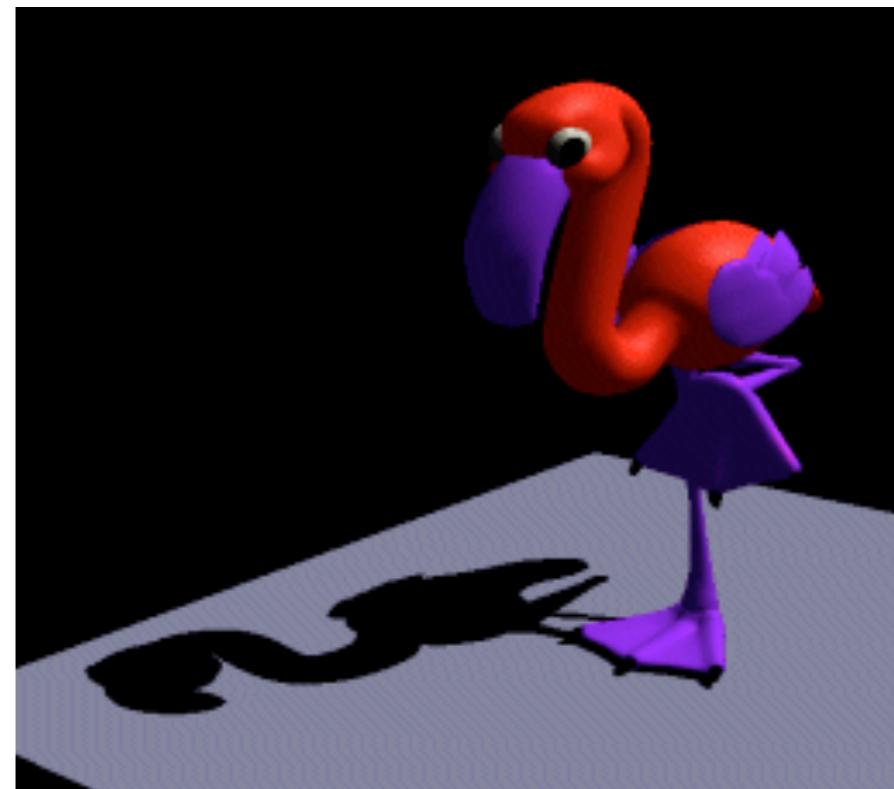
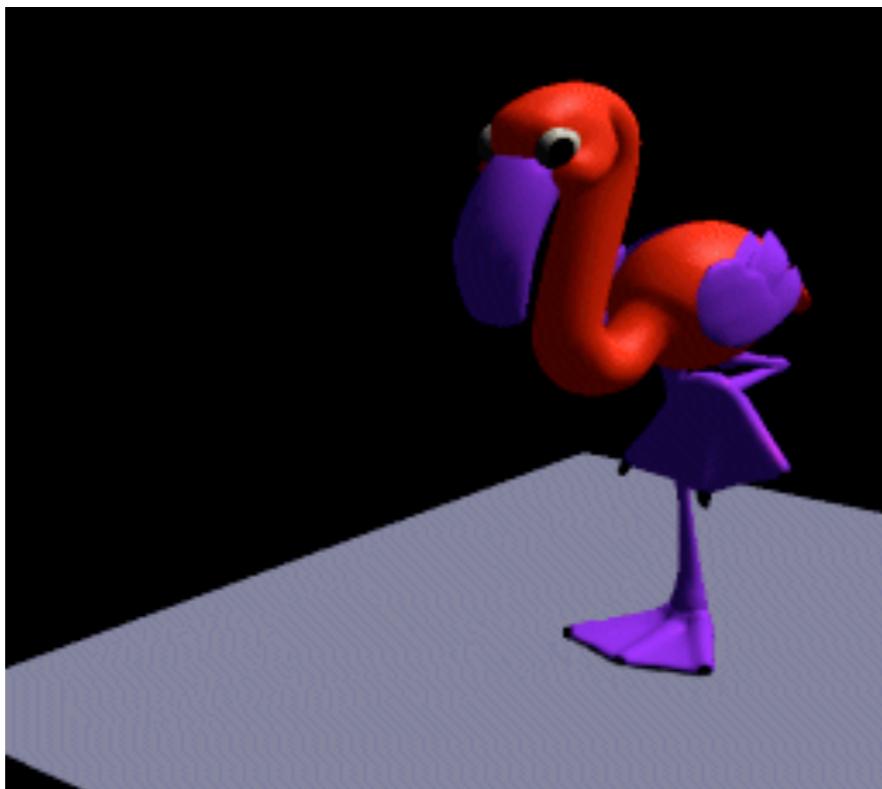
- **Per què ens serveixen lesombres?**
 - Ens donen més sensació de 3D
 - Donen pistes sobre les formes i el posicionament dels objectes en 3D



2.4. Ombres

- **Per què ens serveixen lesombres?**

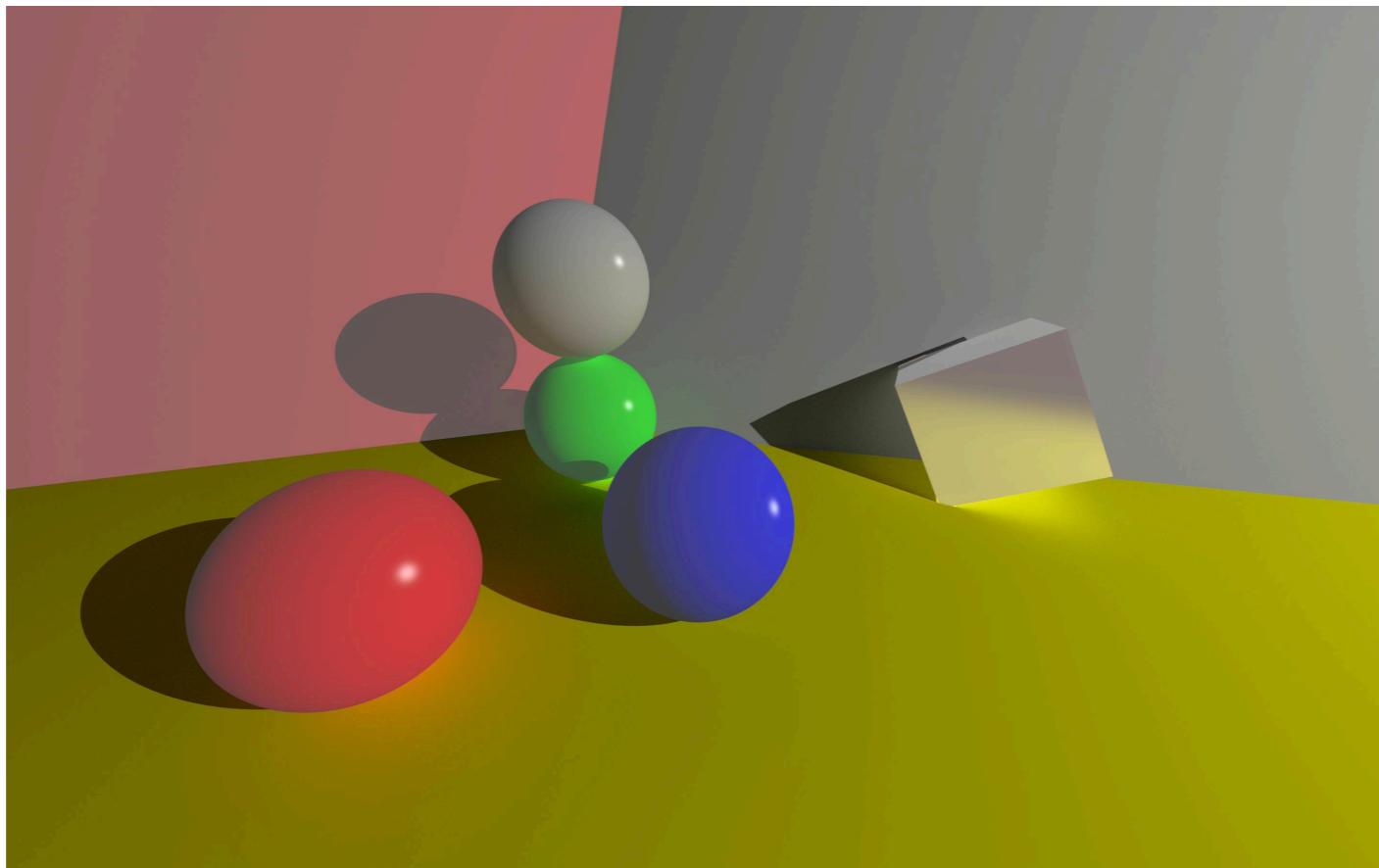
- Ens donen més sensació de 3D
- Donen pistes sobre les formes i el posicionament dels objectes en 3D



2.4. Ombres

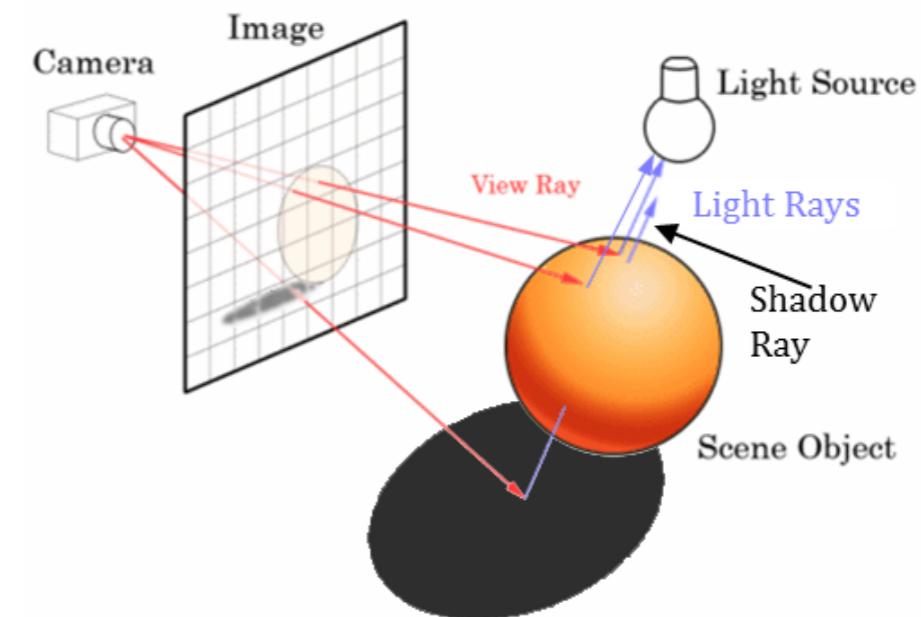
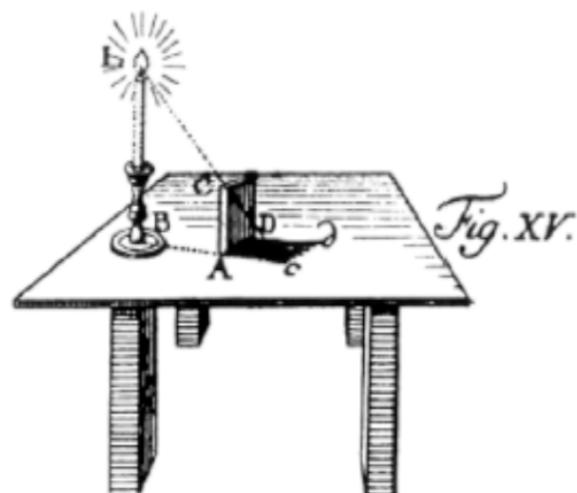
- **Què són lesombres?**

Regions que no són visibles des de les fonts de llum



2.4. Ombres

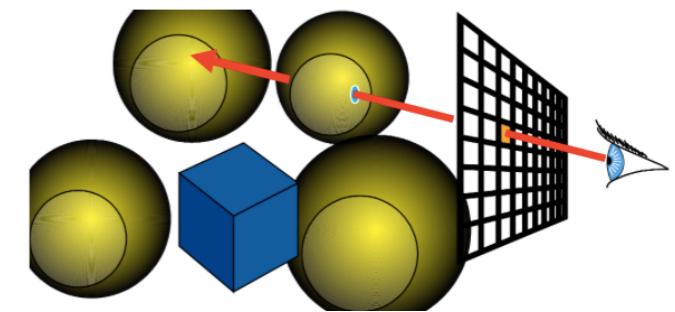
- **Rajos d'ombres:** rajos des del punt d'intersecció a les llums



- Com s'integra en el RayTracing?

2.4. Ombres

- **Raycast() // RayTracer.cpp**
per cada pixel x,y
 raig = scene->getRay(x, y)
 color(pixel) = **RayPixel (raig) // (computeColor(raig))**
 posaColorPixel(x, y, color)
- **RayPixel(ray) { // RayTracer.cpp**
 intersectInfo= **hit(ray, tmin, tmax, hitInfo);**
 si intersecta retorna **Shade(scene, hitInfo, lookFrom)**
 sino retorna Background_Color // o Intensitat ambient global
}
- **Shade(scene, hitInfo, lookFrom) // ShadingStrategy:** retorna el color en el punt intersecció
 color = blinnPhong(point, normal, material, lights)
 retorna color usant fórmula de Blinn-Phong



2.4. Ombres

- Blinn-Phong modificat:

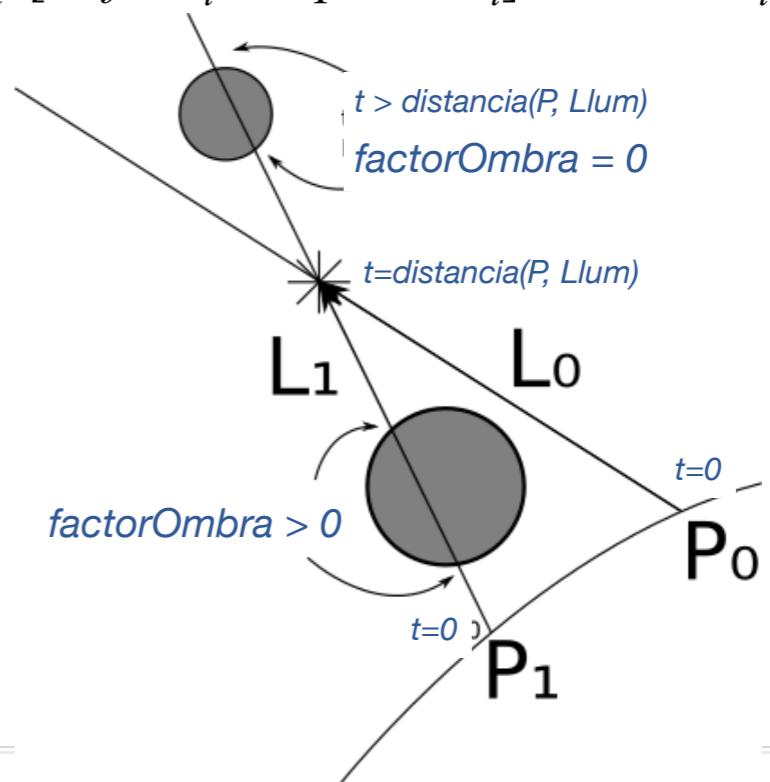
blinnPhong (point, normal, material, llums) // retorna la il·luminació en el punt
retorna color usant fórmula de Blinn-Phong

point conté el punt d'intersecció,
normal és la normal en el punt d'intersecció i
material és el material de l'objecte on està el punt d'intersecció

$$I_{total} = I_{a_global} * K_a + \sum_{i=1}^{numLLums} factor_Ombra_i \cdot atenuació_i \cdot [difusa_i + especular_i] + ambient_i$$

factorOmbra és 0.0 si la llum està totalment
ocluïda per un objecte opac

factorOmbra és 1.0 si no hi ha cap objecte
entre la llum i el punt p



2.4. Ombres

blinnPhong (`point`, `normal`, `material`, `llums`) **retorna color**

// retorna la il·luminació en el punt tenint en compte ombres

`c = lambientGlobal * ka`

per a cada llum

`rL = càcul raig de llum entre point i la llum`

`factorOmbra = calculOmbra(rL);`

`c += ambient + factorOmbra * (difús + especular);`

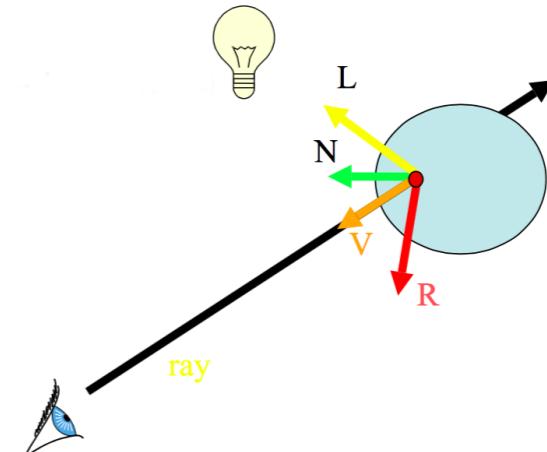
fper

retorna c

Cal calcular si hi ha ombra?
factorOmbra està entre 0 i 1

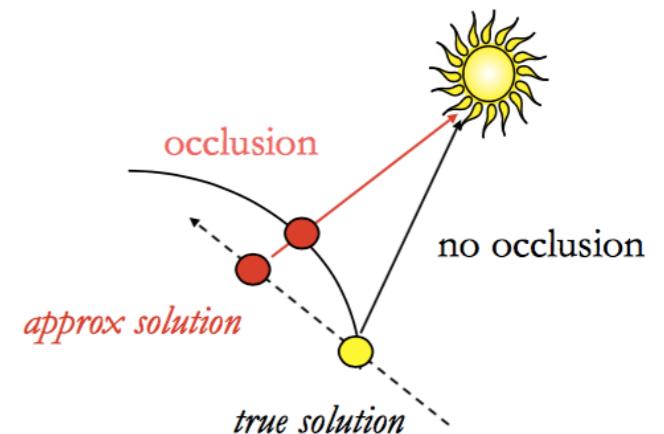
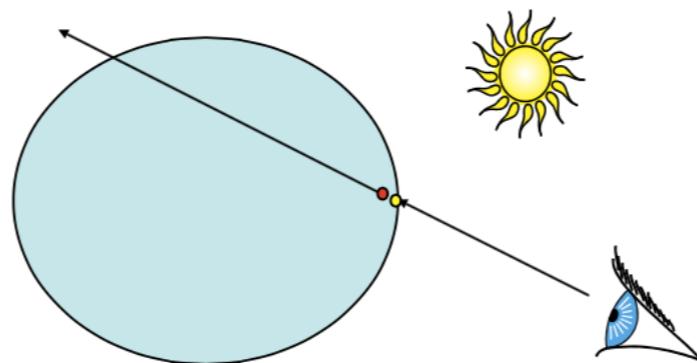
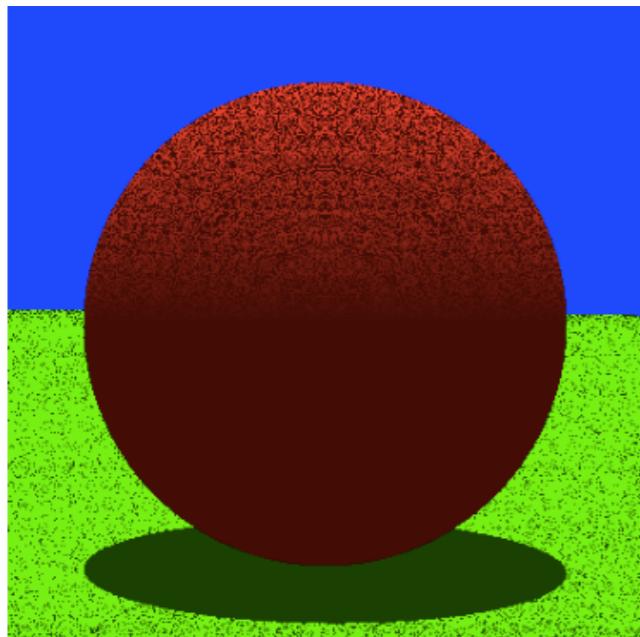
factorOmbra és 0.0 si la llum està totalment
ocluïda per un objecte opac

factorOmbra és 1.0 si no hi ha cap objecte
entre la llum i el punt p



2.4. Ombres

- **Problema de les auto-interseccions (self-shadowing):**
Problema de precisió (shadow acne)

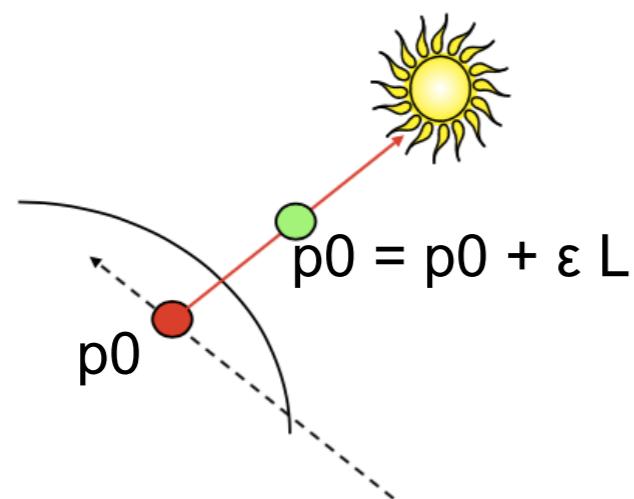


Rraig cap a la llum es defineix com:

$$p = p_0 + \lambda * L$$

On,

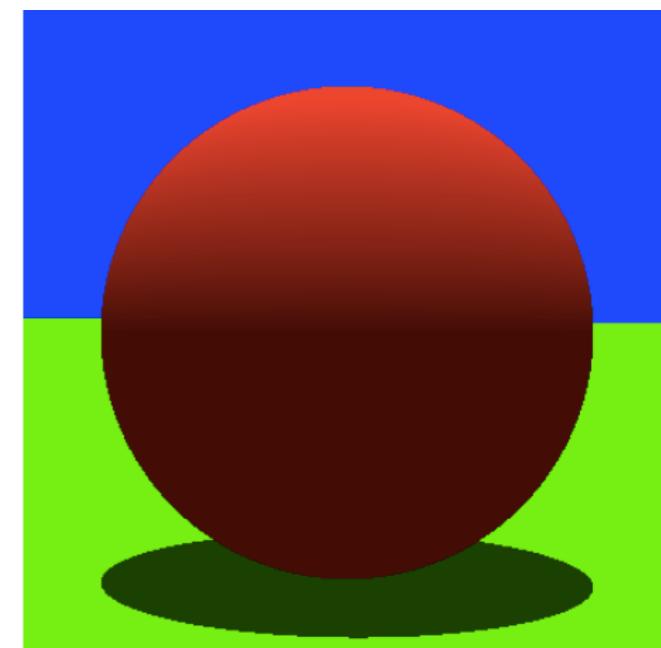
p_0 és el punt d'intersecció,
 L és el vector de p_0 a la
posició de la llum



Es canvia p_0 per:

$$p_0 = p_0 + \epsilon * L,$$

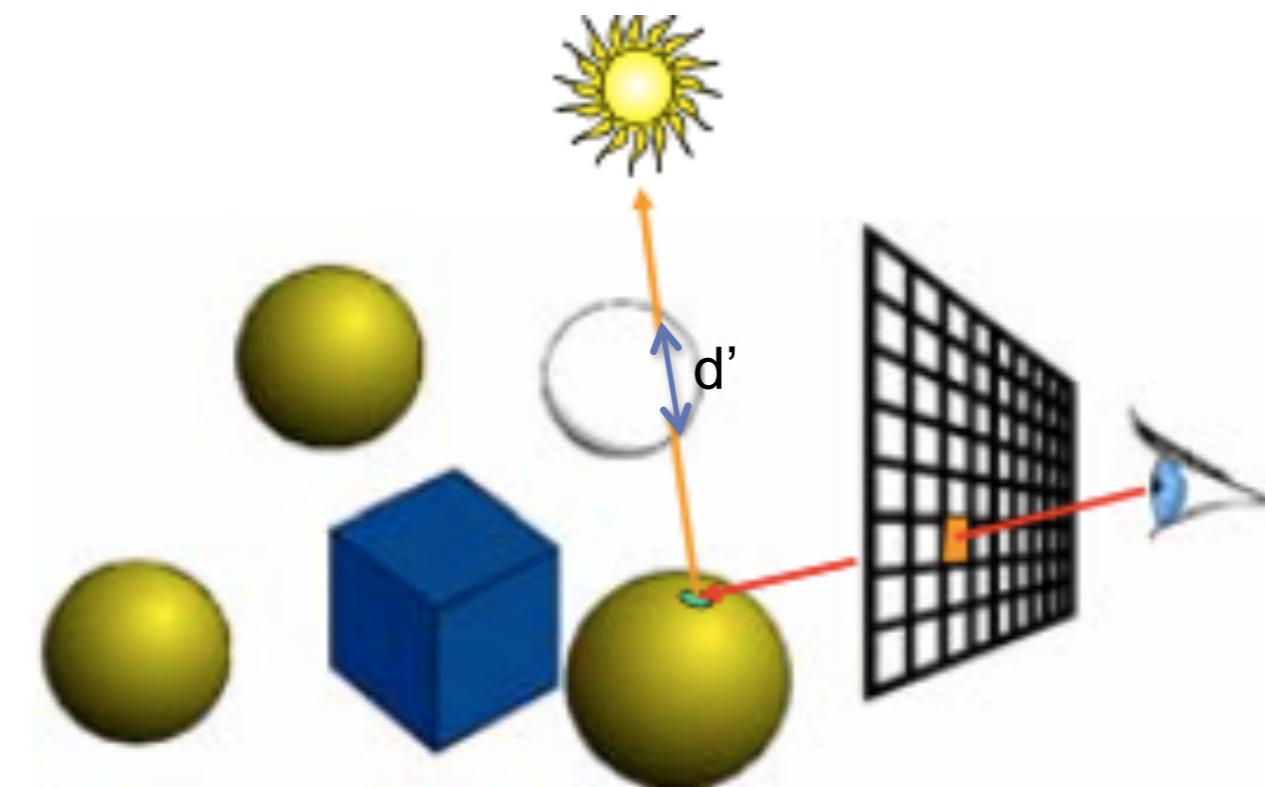
amb $\epsilon = 0.01$



2.4. Ombres

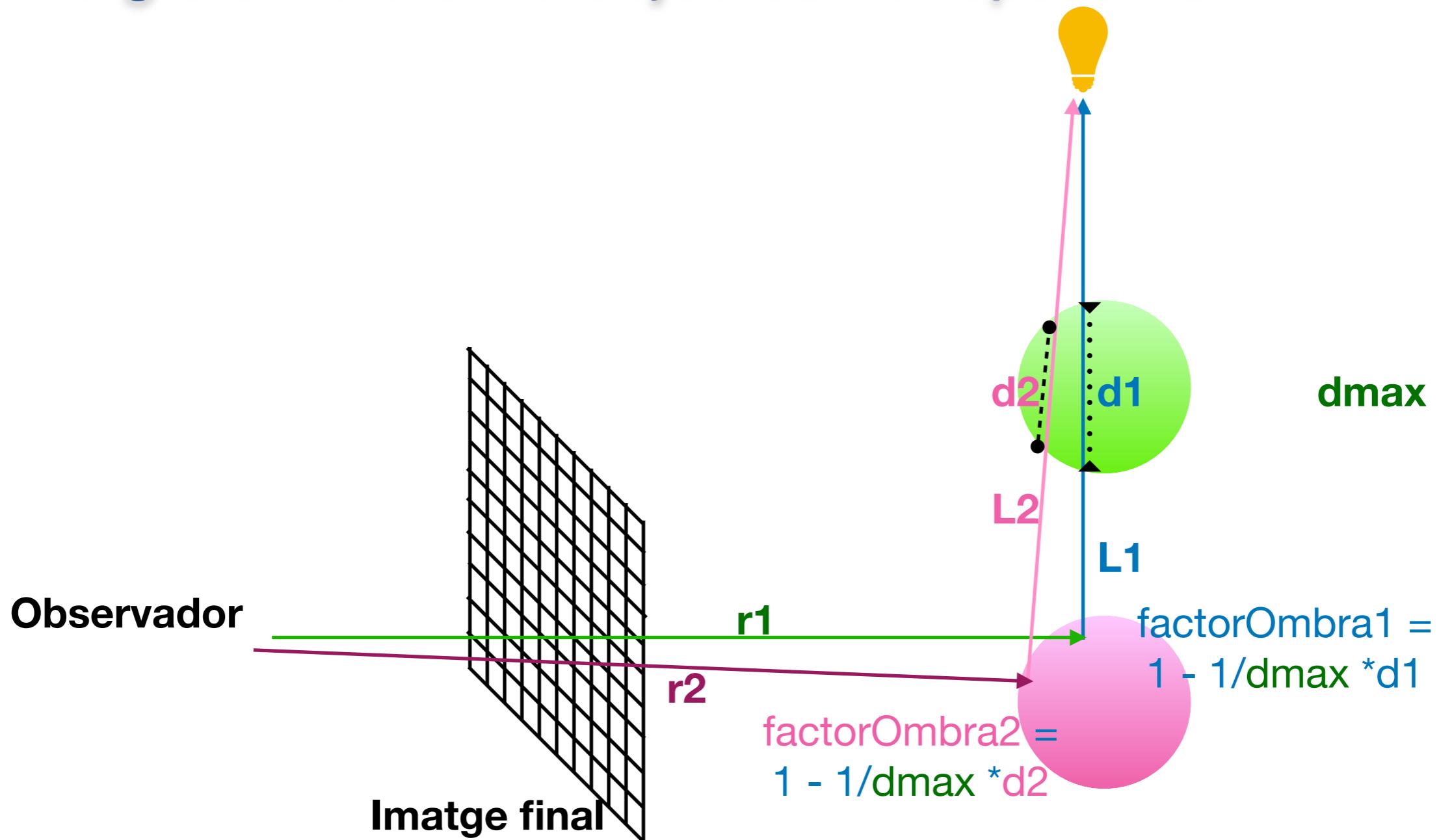
- **Rraig d'ombra amb objectes transparents:**

- Es tracta com si fos transparent, però sense tenir en compte la refracció del raig de la llum:



2.4. Ombres

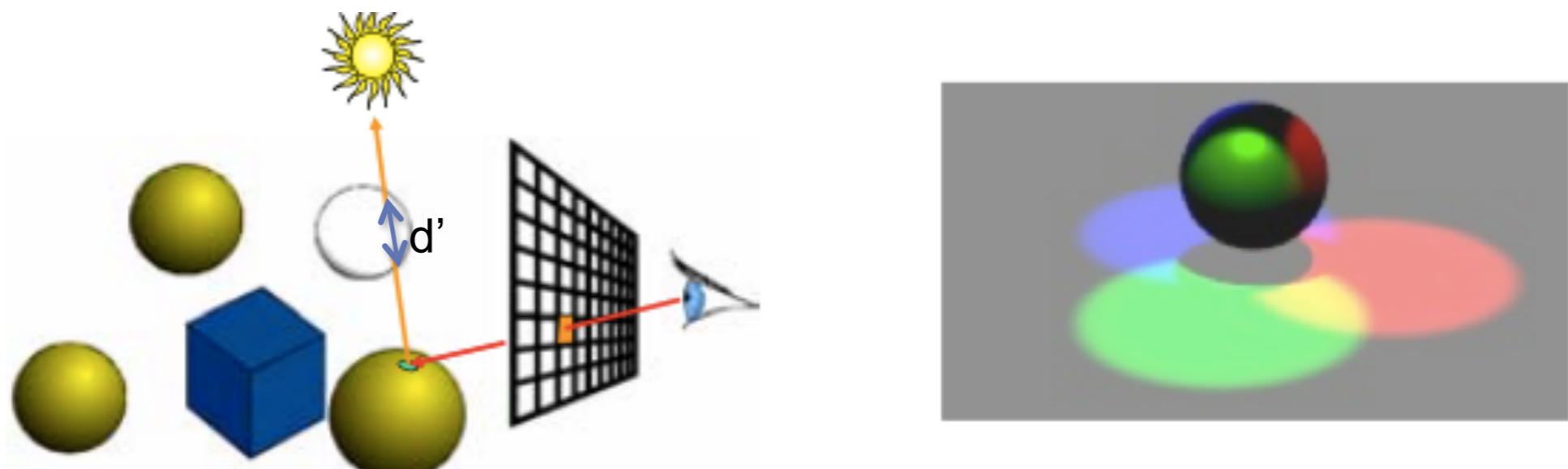
- Raig d'ombra amb objectes transparents:



2.4. Ombres

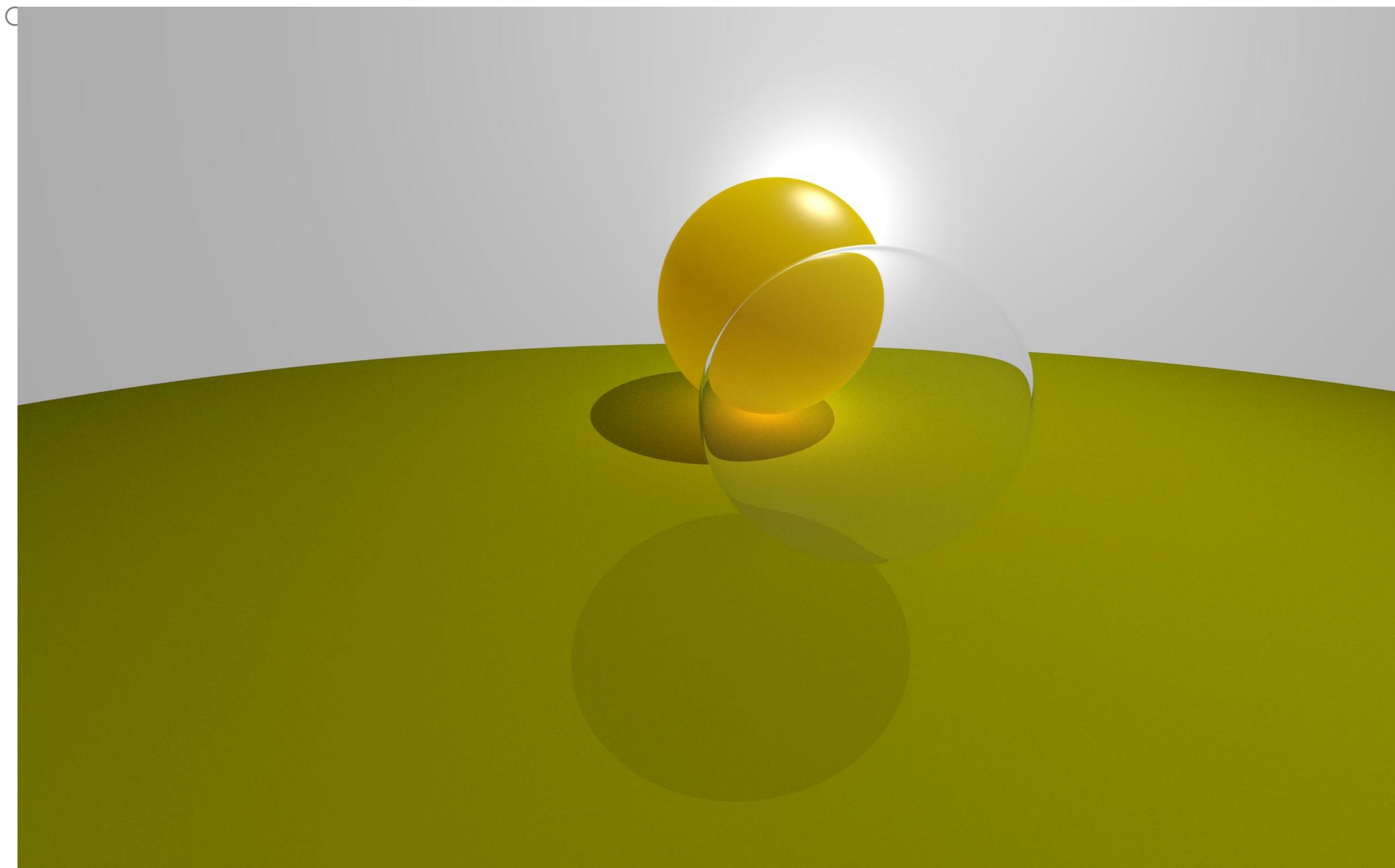
- **Rraig d'ombra amb objectes transparents:**

- Es calcula el gruix de l'objecte transparent que travessa el rraig d'ombra (amb les dues interseccions del rraig amb l'objecte): **d'**.
- Es defineix un factor d'atenuació en el material transparent (**fah** = 1.0 / **dmax**, on **dmax** és el gruix de l'objecte que faria que la llum no passés)
- Es calcula factor d'ombra **factorOmbra** = $(1.0 - \text{fah} * \mathbf{d'})$ que s'aplicarà a la component difusa directa i la component specular directa de Blinn-Phong



2.4. Ombres

- Raig d'ombra amb objectes transparents:

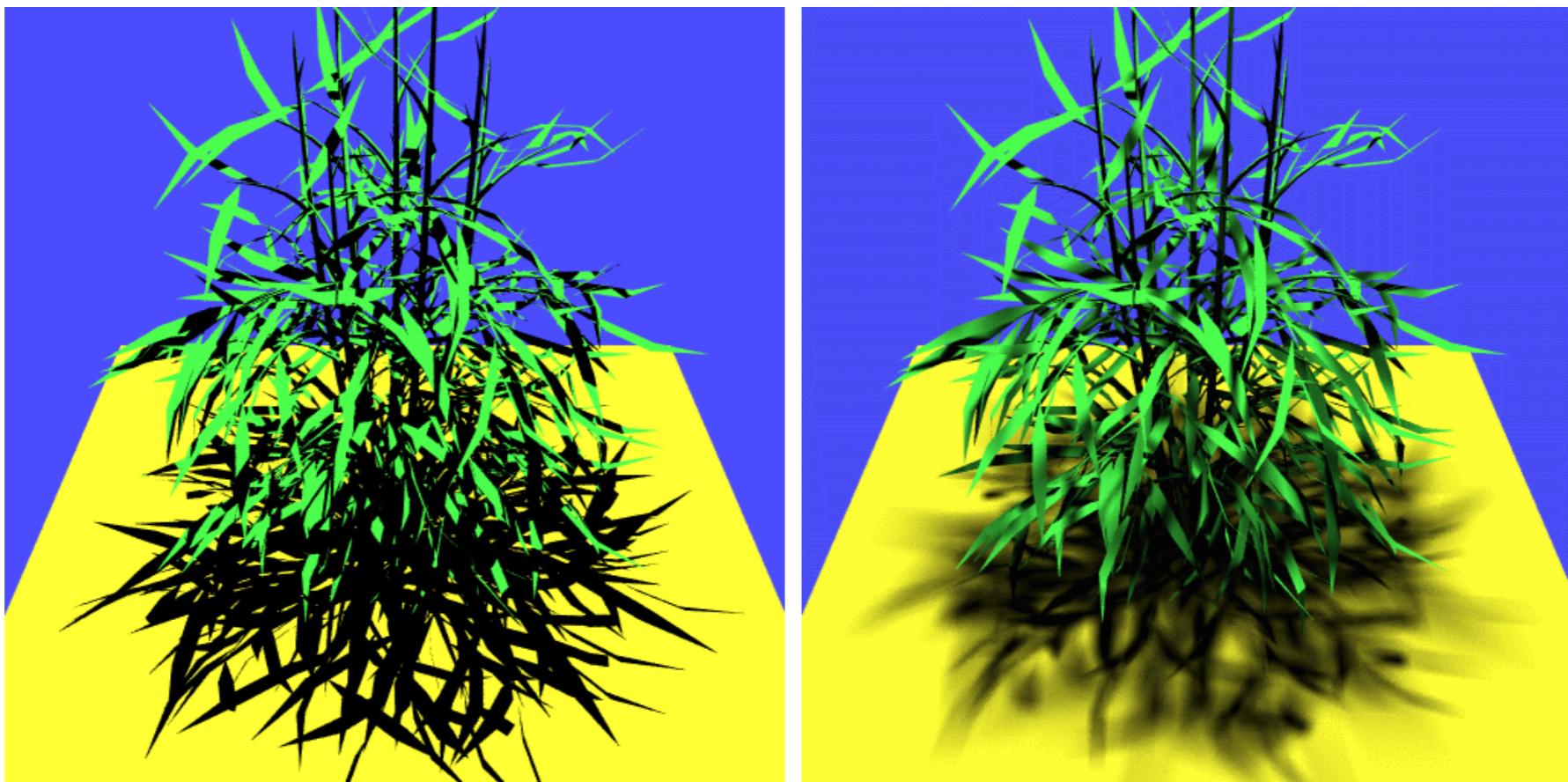


2.4. Ombres

- **Soft Shadows (ombres i penombres):**

Les llums **puntuals** produeixen ombres molt dures

Si es considera que les llums tenen àrea i no són punts, es produeixen les **penombres**



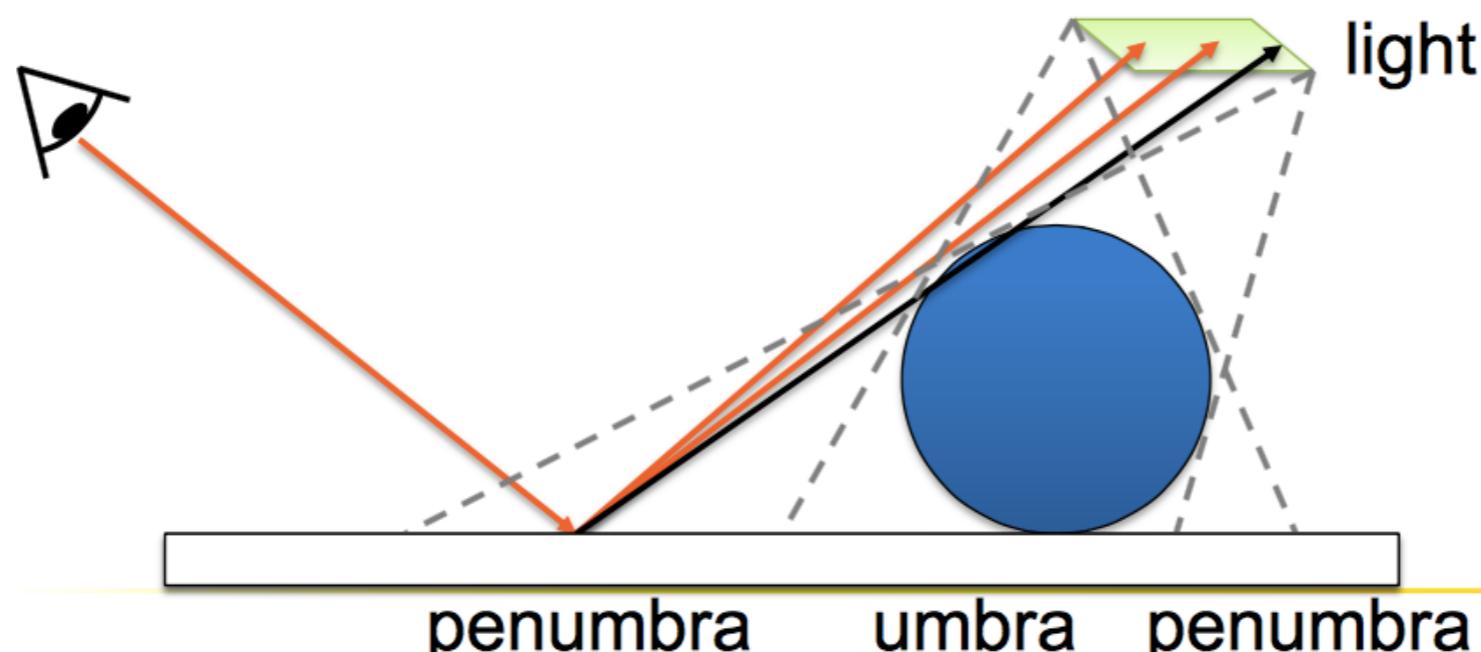
2.4. Ombres

- **Soft Shadows (penombres):**

Les llums amb àrea fanombres suaus per què:

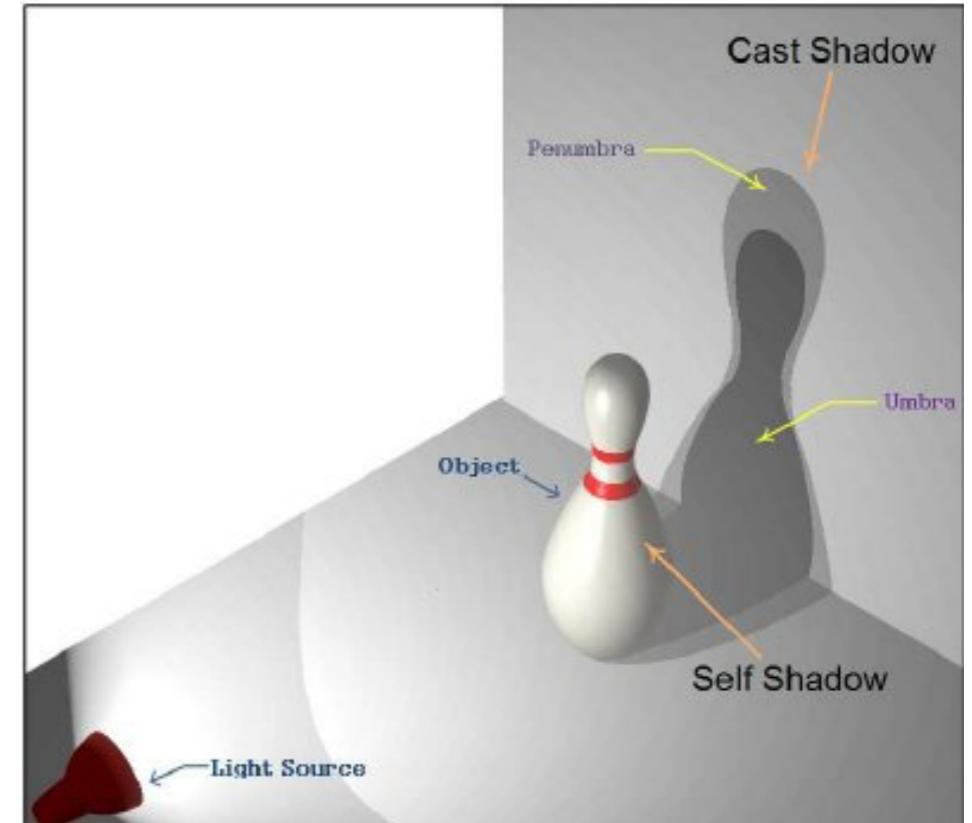
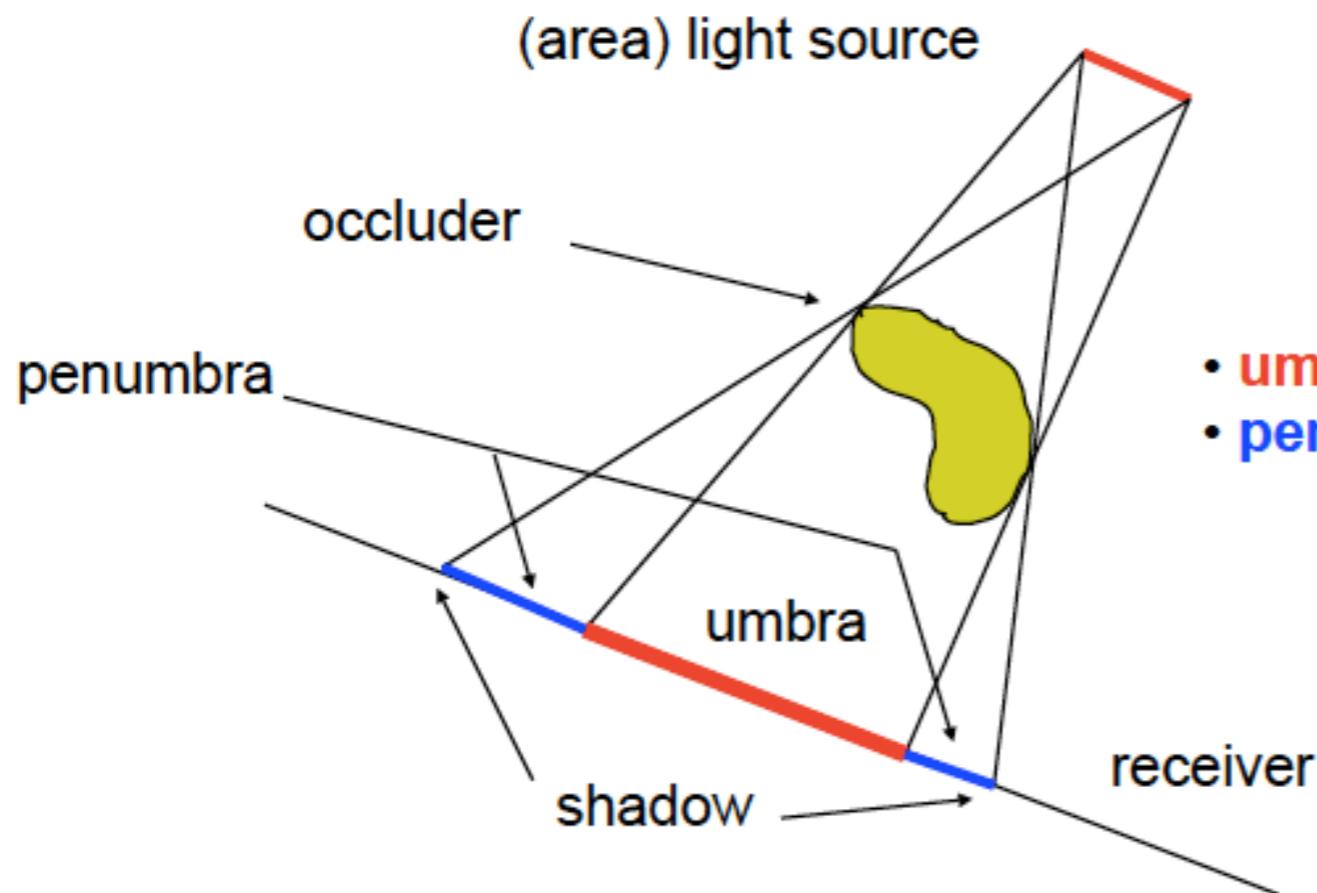
- alguns punts “veuen” tota la llum - **totalment il·luminats**
- alguns punts no “veuen” res de la llum - **ombra**
- alguns punts veuen part de la llum - **penombra**

Es llencen varis rajos d'ombra des d'un mateix punt i es pondera l factor d'ombra segons el nombre de rajos que veuen la llum.



2.4. Ombres

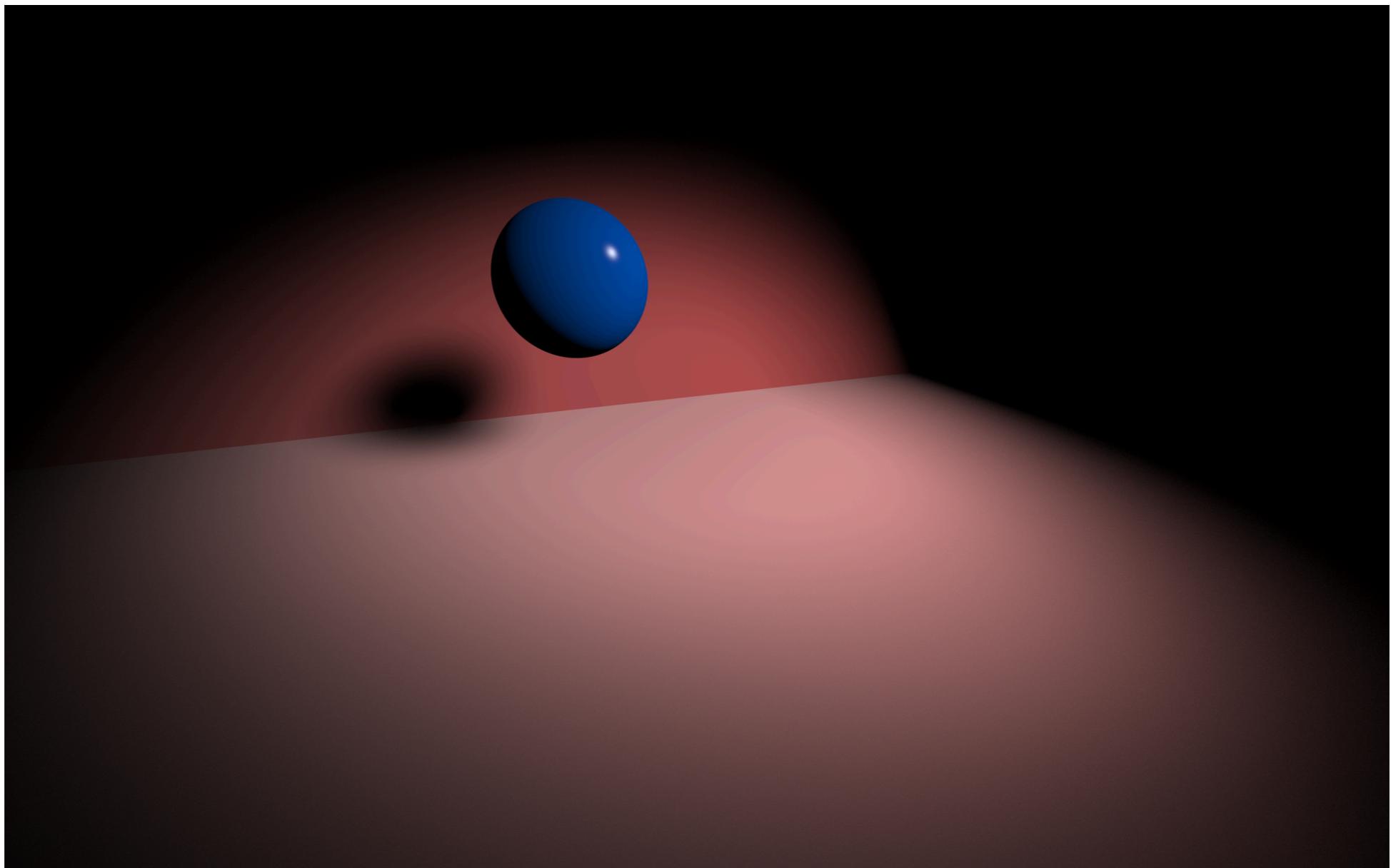
- Soft Shadows (penombres):



- **umbra** – fully shadowed region
- **penumbra** – partially shadowed region

2.4. Ombres

- Soft Shadows (penombres):



Pere Gilabert i Adrià Torralba