

Iluminación

Informática Gráfica I

Material de: **Antonio Gavilanes y Ana Gil**
Adaptado por: **Elena Gómez y Rubén Rubio**
{mariaelena.gomez,rubenrub}@ucm.es



Contenido

1 Introducción

- Luz y color
- Renderización

2 Modelos de iluminación

- Modelo de Phong
- Componente difusa
- Componente especular
- Componente ambiente

- Luz final

3 Fuentes de Luz

- Tipos
- Shading models

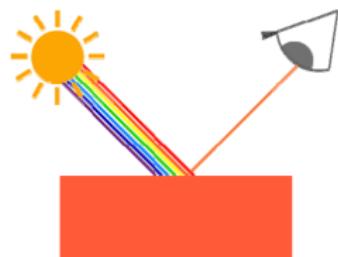
4 Materiales

5 Iluminación global

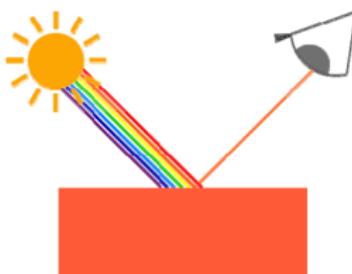
- Tipos
- Sombras, reflejos y cáusticas

Luz y color

- El **color con el que vemos un objeto** depende de la luz que incide en la superficie: Parte de la luz es reflejada y parte es absorbida por el objeto.
- El color que percibimos son los colores reflejados.
- El color del objeto es el que percibimos con luz blanca: (1, 1, 1)



Luz y color



```
vec3 lightColor = {1.0, 1.0, 1.0};  
vec3 objColor = {1.0, 0.5, 0.31};  
vec3 result = lightColor * objColor; // = {1.0, 0.5, 0.31};  
vec3 lightColor = {0.0, 1.0, 0.0};  
vec3 objColor = {1.0, 0.5, 0.31};  
vec3 result = lightColor * objColor; // = {0.0, 0.5, 0.0};
```

Luz y color

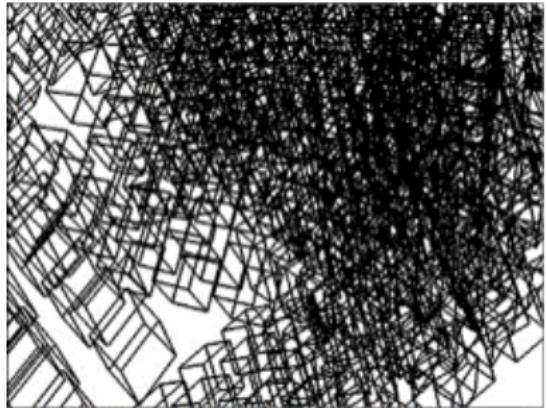
- El color con el que vemos un punto de un objeto se define en función de:
 - La luz que incide en el punto:
 - Color e intensidad de la luz (RGBA)
 - Dirección de incidencia (vector)
 - Las propiedades del material (reflectancia):
 - Color (RGBA): porcentaje de la luz incidente que será reflejada (coeficiente de reflexión)
 - Brillo: La forma en que se refleja
 - Del punto de vista (cámara), pues la luz se refleja:
 - Uniformemente en todas las direcciones
 - En dirección especular con la de incidencia



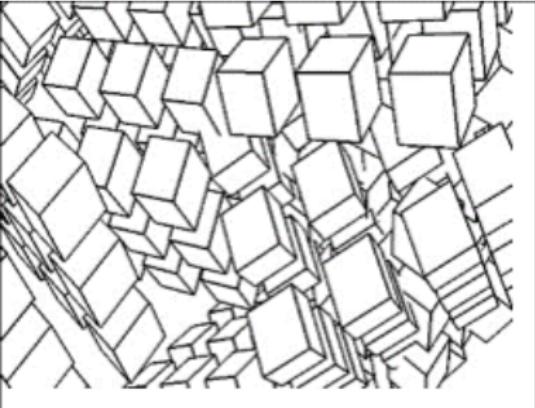
Renderización

- Renderización de escenas
- Jerarquía en los niveles de realismo

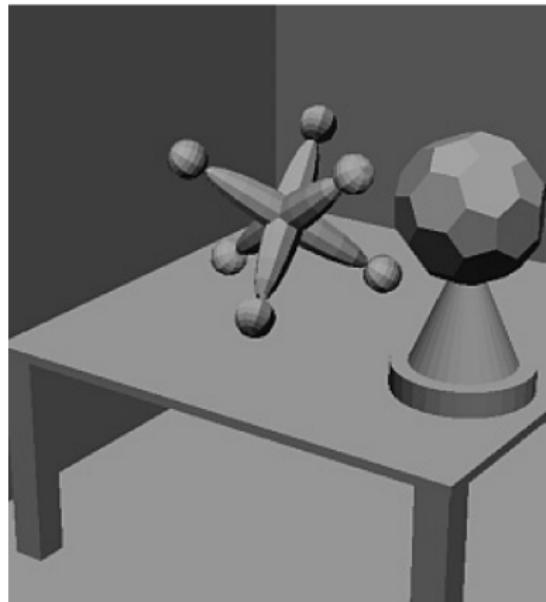
Renderización basada
en armazón de hilos



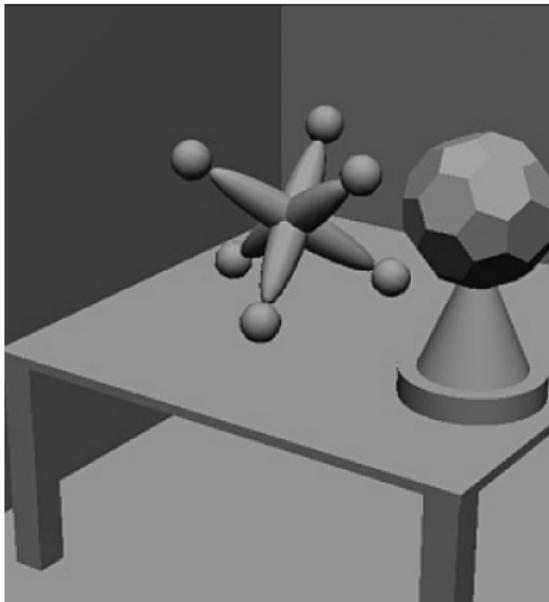
Rend. basada en armazón
de hilos con eliminación
de superficies ocultas



Renderización

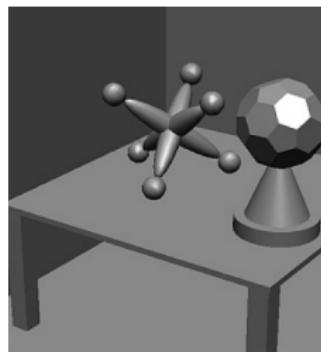


Iluminación plana
(*flat shading*)

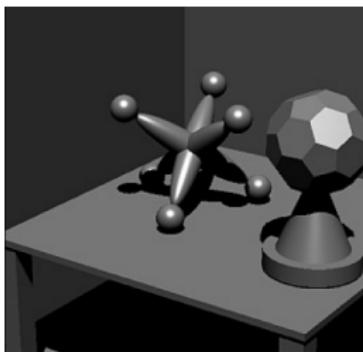


Iluminación suave
(*smooth shading*)

Renderización



Iluminación con
efectos
especulares



Iluminación
con sombras



Iluminación
con texturas

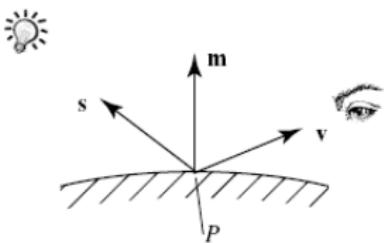
Modelos de iluminación

- Un modelo de iluminación (*lighting model*) dicta cómo se refleja y dispersa la luz, cuando incide sobre una superficie.
- Se define como el cálculo de la intensidad de cada punto de la escena.
- Es una simplificación con respecto al modelo físico de propagación de la luz (absorción de luz, transmisión de la luz a través de los objetos, intensidad de la luz emitida).
- En dicho cálculo intervienen:
 - El tipo e intensidad de las fuentes de luz.
 - El material del objeto.
 - La orientación del objeto con respecto a la luz.
- Fenómenos principales que se tienen en cuenta cuando la luz incide en una superficie:
 - La **dispersión difusa**, responsable del color de la superficie.
 - El **reflejo especular**, responsable del material de la superficie.

Modelos de iluminación

- Elementos geométricos:

- **Vector** normal a la superficie en el punto P (m).
- **Vector** desde P al ojo de la cámara (v).
- **Vector** desde P a la fuente de luz (s).



Modelo de iluminación de Phong

- La luz reflejada por una superficie consta de una combinación de tres componentes: **difusa**, **especular** y **ambiente**.
 - Las componentes difusa y ambiente representan el color del material.
 - La componente especular representa los reflejos del material.

Modelo de iluminación de Phong

- La luz reflejada por una superficie consta de una combinación de tres componentes: **difusa**, **especular** y **ambiente**.
- Tanto el color y la intensidad de la luz, como la reflectancia del material se definen para cada una de las tres componentes, mediante un valor RGBA:

diffLightColor

specLightColor

ambLightColor

diffMaterialColor

specMaterialColor

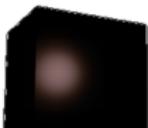
ambMaterialColor



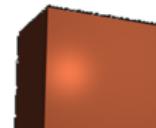
ambiente



difusa



especular

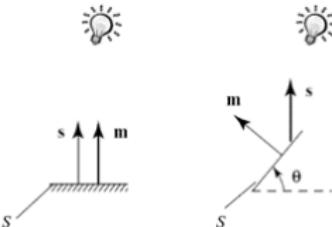
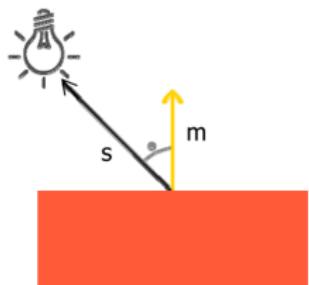


combinadas

Componente difusa

Dispersión difusa

- Luz que, habiendo incidido sobre una superficie, se vuelve a irradiar desde ella, en todas las direcciones (independiente del punto de vista)
- Su intensidad depende del ángulo θ que forman los vectores s y m .



Para fuentes de luz lejanas, el vector S varía poco, luego la componente difusa cambia poco a lo largo de la superficie.

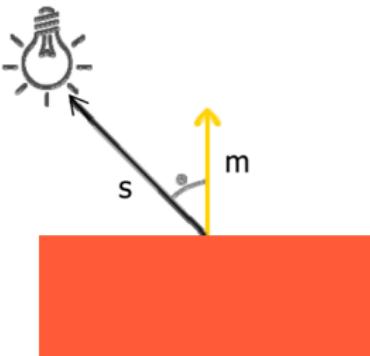
Componente difusa

Dispersión difusa

- Para su implementación se usa el modelo de **Lambert**:

```
vec3 diffuse = max(0, cos(theta)) * diffLight * diffMaterial
```

donde $\cos \theta = m \cdot s = \text{dot}(m, s)$ (si los vectores están normalizados).



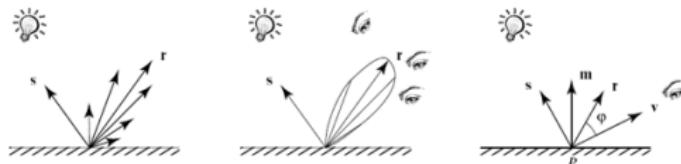
Componente especular

Reflexión especular

- Luz refleja la superficie
- El vector de máxima reflexión $r = \text{reflect}(s, m)$ coincide con el rayo reflejado. La intensidad de la luz reflejada depende pues del ángulo que forman los vectores r y v .
- Para su implementación se usa el modelo de **Phong**:

```
vec3 specular = max(0, pow(cos(theta), f)) *
    specLight * specMaterial
```

donde $\cos \theta = r \cdot v$ (si los vectores están normalizados) y $f = 1, \dots, \infty$ (espejo) es un atributo del material.



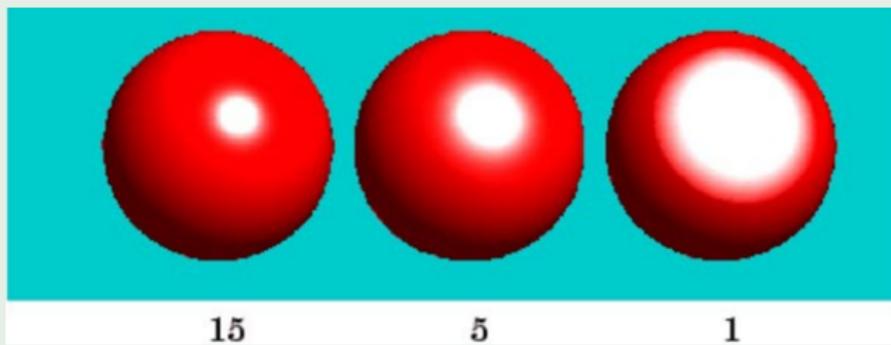
Para fuentes de luz lejanas, los vectores r y s varían mucho, luego la componente especular cambia mucho a lo largo de la superficie.

Componente especular

Reflexión especular

```
vec3 specular = max(0, pow(cos(theta), f))  
    * specLight * specMaterial
```

Valores de f



Componente ambiente

Ambiente de las fuentes de luz

- Luz que rebota en numerosas superficies de la escena.
- Luz que alcanza una superficie aunque no esté expuesta a la fuente de luz.
- Es una luz que ilumina a todos los objetos por igual.
- Se implementa siguiendo un modelo simple:

```
vec3 ambient = ambLight * ambMaterial
```

Resultado final

- El color resultante de una fuente de luz es la combinación de las tres componentes obtenidas:

```
vec3 finalColor = diffuse + specular + ambient
```

- En una escena pueden haber varias fuentes de luz.
- El color final de un punto de la superficie de un objeto es la suma de los colores finales por cada fuente de luz, teniendo en cuenta además otros factores, como la atenuación de las fuentes, la emisividad del material, la pertenencia a conos de luz.

El shader simple_light

```
void main() // fragment shader: uniform lightDir, color;  
{           // in FragPos, Normal; out FragColor  
    vec3 normal = normalize(Normal); // por la interpolación  
  
    // Componente difusa  
    float diff_factor = max(dot(normal, - lightDir.xyz), 0.0);  
    // Componente especular (con coeficiente de Phong)  
    vec3 viewDir = normalize(- FragPos); // dirección de la vista  
    vec3 reflectDir = reflect(lightDir.xyz, normal);  
    float spec_factor = pow(max(dot(viewDir, reflectDir), 0.0),  
                           specularExponent);  
  
    FragColor = color * (ambient + diff_factor * diffuse  
                        + spec_factor * specular);  
}
```

Fuentes de luz en OpenGL

Las fuentes de luz tradicionalmente consideradas en OpenGL son de dos tipos:

- **Direccionales** (o remotas): se supone que están infinitamente lejos y que, por tanto, sus rayos de luz llegan paralelos a la escena.
 - Por su lejanía infinita se supone que no se atenúan.
 - Se especifican mediante el vector que va del origen a la fuente de luz.

Fuentes de luz en OpenGL

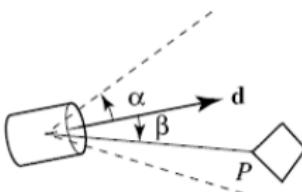
- **Posicionales** (o locales): Se supone que están localizadas en un punto y que, por tanto, los objetos de la escena son más o menos iluminados según su exposición.
 - Se especifican mediante el punto en el que se encuentra la fuente de luz. En estos casos, el vector `s` en un punto del objeto se obtiene por
$$s = (\text{lightPosition} - \text{pointPosition}).normalize()$$
.
 - Se puede conocer también la distancia de la fuente al punto por lo que estas fuentes admiten atenuación.

Fuentes de luz en OpenGL

En OpenGL se suelen considerar dos tipos de fuentes de luz **posicionales**:

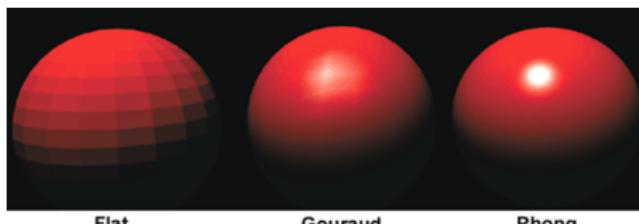
- **Omnidireccionales**: emiten luz por igual en todas direcciones.
- **Focos**: se caracterizan por una dirección (el vector d) y una amplitud (el ángulo α) de emisión. Los puntos que se encuentran fuera del cono de emisión de luz que determinan estos dos parámetros no reciben luz del foco.

Los que se encuentran dentro del cono de emisión reciben una cantidad de luz que varía según el ángulo β en un factor que es una potencia ϵ del $\cos \beta$. Cuanto mayor es ϵ , más se concentra la luz del foco.



Shading models

- Modelo de matizado o tonalidad (shading model)
 - **Flat shading**: todos los fragmentos del interior de un triángulo se colorean igual. Los tres vértices del triángulo tienen el mismo vector normal: la **normal al triángulo**.
 - **Gouraud shading**: los vértices del triángulo tiene distinto vector normal: **la normal a la superficie que aproxima**. Se calcula el color para cada uno de los tres vértices y el color de cada fragmento se obtiene por interpolación de los colores de los vértices.
 - **Phong shading**: los vértices del triángulo tiene distinto vector normal, el calculo del color se realiza para cada fragmento.



Flat

Gouraud

Phong

Shading models en OpenGL

- En versiones antiguas de OpenGL, el shading model se escogía con una función `glShadeModel` de la API.
- En el *core profile*, los modelos de sombreado se corresponden con:
 - **Gouraud**: el cálculo de la iluminación se implementa en el shader de vértices.
 - **Pong**: el cálculo de la iluminación se implementa en el shader de fragmentos.
 - **Flat**: en ambos casos, se puede conseguir desactivando la interpolación de normales o colores con el *interpolation qualifier* `flat` la variable de entrada o salida del shader.

Materiales

- OpenGL simula **materiales** por la forma en que reaccionan ante la luz roja, verde y azul en cada componente (difusa, especular, ambiente) de la fuente de luz.
Por ejemplo, en cuanto al color, una superficie con componentes difusa y ambiente verde $(0, 1, 0)$ bajo una luz con componentes difusa y ambiente roja $(1, 0, 0)$ resulta ser $(0 \times 1, 1 \times 0, 0 \times 0) = (0, 0, 0)$.
- En consecuencia, en OpenGL, los materiales se especifican con las mismas tres componentes (ambiente, difusa y especular) que las fuentes de luz.

Materiales

- En el caso de los materiales, las componentes se llaman **coeficientes de reflexión** o **reflectancias** del material.
- Cada coeficiente del material indica cómo reacciona la superficie ante la correspondiente componente de cada fuente de luz.
Por ejemplo, una superficie con coeficiente de reflexión especular alto parecerá brillante bajo una fuente de luz con componente especular alta.

Materiales

- Como los coeficientes de reflexión ambiente y difuso son los responsables del color, y el color de una entidad no cambia se enfrente a una luz o no, en los materiales estos coeficientes se suelen tomar iguales
- Como el coeficiente especular solo es responsable de los reflejos que tiene la superficie, en los materiales este coeficiente se suele tomar gris de forma que el color de la superficie no se vea alterado por el color de la luz incidente.

Materiales

- Coeficientes de reflexión para simular materiales comunes (junto con el valor del exponente para usarse en la componente especular).

Material	ambient $\rho_a = \rho_{ag} \rho_{ab}$	diffuse $\rho_d = \rho_{dg} \rho_{db}$	specular $\rho_s = \rho_{sg} \rho_{sb}$	exponent f
Black Plastic	0.0	0.01	0.50	32
	0.0	0.01	0.50	
	0.0	0.01	0.50	
Brass	0.329412	0.780392	0.992157	27.8974
	0.223529	0.568627	0.941176	
	0.027451	0.113725	0.807843	
Bronze	0.2125	0.714	0.393548	25.6
	0.1275	0.4284	0.271906	
	0.054	0.18144	0.166721	
Chrome	0.25	0.4	0.774597	76.8
	0.25	0.4	0.774597	
	0.25	0.4	0.774597	
Copper	0.19125	0.7038	0.256777	12.8
	0.0735	0.27048	0.137622	
	0.0225	0.0828	0.086014	
Gold	0.24725	0.75164	0.628281	51.2
	0.1995	0.60648	0.555802	
	0.0745	0.22648	0.366065	
Pewter	0.10588	0.427451	0.3333	9.84615
	0.058824	0.470588	0.3333	
	0.113725	0.541176	0.521569	
Silver	0.19225	0.50754	0.508273	51.2
	0.19225	0.50754	0.508273	
	0.19225	0.50754	0.508273	
Polished Silver	0.23125	0.2775	0.773911	89.6
	0.23125	0.2775	0.773911	
	0.23125	0.2775	0.773911	

Iluminación global

- Técnicas de iluminación que tienen en cuenta las interacciones entre los distintos objetos de la escena.
- Considera la luz reflejada un punto teniendo en cuenta toda la luz que llega y no solo la energía que proviene de las fuentes de luz.
- Apariencia visual más realista.
- Coste computacional alto.

Modelos de iluminación locales vs globales

● Modelo local

- El color de un vértice depende del material y de las fuentes de luz.
- No se tienen en cuenta sombras ni luces reflejadas.
- Solo interacción objeto-fuentes de luz.
- Fenómenos ópticos comunes (sombras, reflejos, causticas) difíciles de reproducir.

● Modelo global

- El color de un vértice depende del material y de la luz que le llegue de las fuentes de luz.
- Interacción objeto-fuentes de luz y también objeto-objeto.
- Fenómenos ópticos comunes no son tan costosos de reproducir.
- Ray tracing, radiosidad.

Tipos de iluminación globales

- Trazador de rayos (*ray tracing*).
- Radiosidad.
- Path tracing.
- Cachés de irradiancia.
- Photon tracing.

Trazador de rayos

- Lanza rayos que intersecan con los distintos objetos de la escena.
- Determina las superficies visibles en la escena trazando rayos desde el observador (cámara) a través del plano de la imagen.
- Aquella intersección más cercana es la que determina el objeto visible.
- Para simular la reflexión y refracción, los objetos que emiten luz también lanzan rayos.

Trazador de rayos



Trazador de rayos



Disney PIXAR

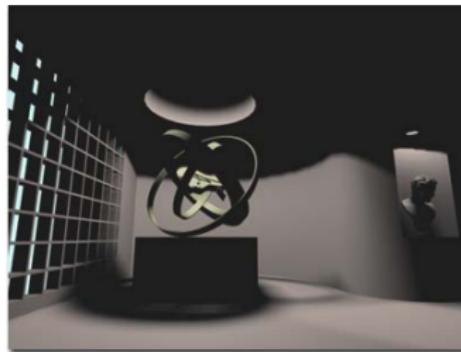
Radiosidad

- Se centra en el intercambio de energía lumínica y en como es absorbida por los objetos.
- Aplicación del método de elementos finitos para resolver la ecuación de renderizado en escenas con superficies puramente difusas.
- Se calcula la cantidad de luz que llega a cada punto de la escena lanzando rayos en todas las direcciones y acumulando la cantidad de luz recibida.
- Se almacena la luz por elemento de la escena.
- Se puede iterar para conseguir más rebotes de luz.

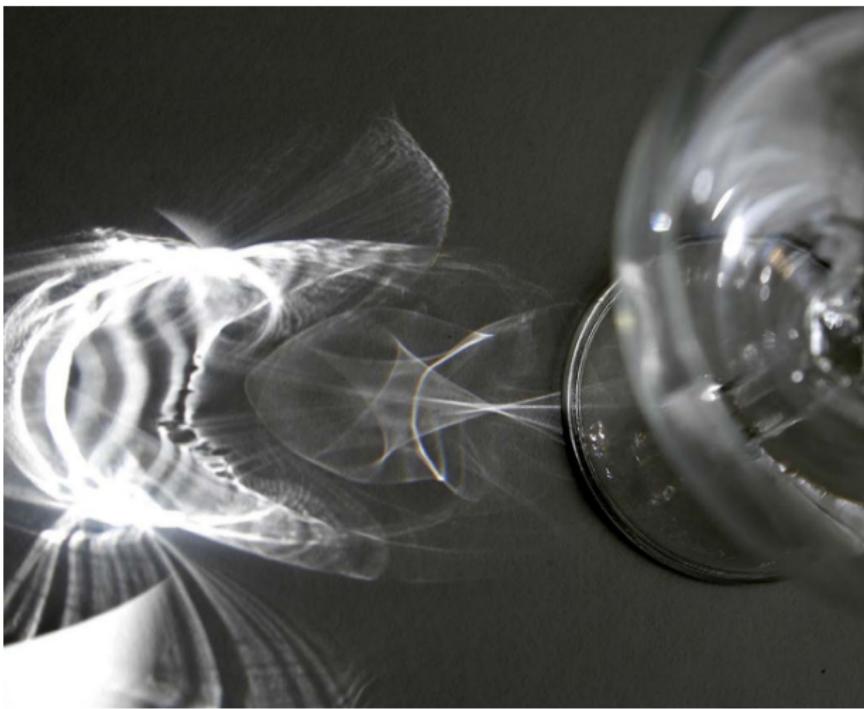
Radiosidad



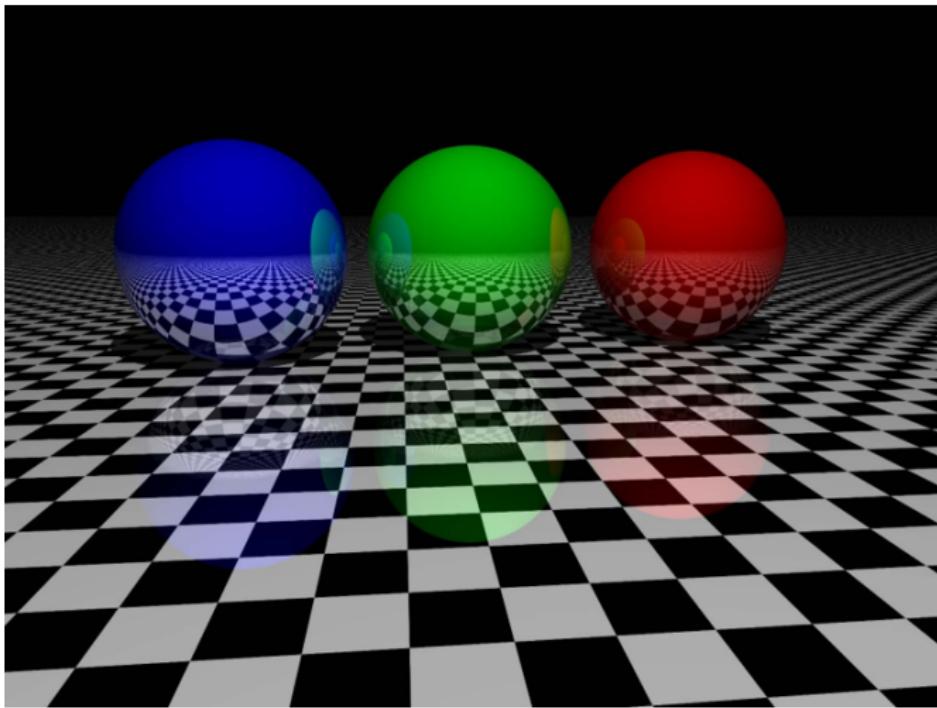
Radiosidad



Cáusticas



Reflejos



Sombras, reflejos y cáusticas

