Tema 5: Modelos de Iluminación y Sombreado

J. Ribelles

SIE020: Síntesis de Imagen y Animación
Institute of New Imaging Technologies, Universitat Jaume I

Contenido

Introducción

≪*Un modelo de iluminación determina el color de la superficie en un punto. Un modelo de sombreado utiliza un modelo de iluminación y especifica cuándo usarlo.*≫

Modelo de Iluminación de Phong

Este modelo tiene en cuenta los tres siguientes aspectos

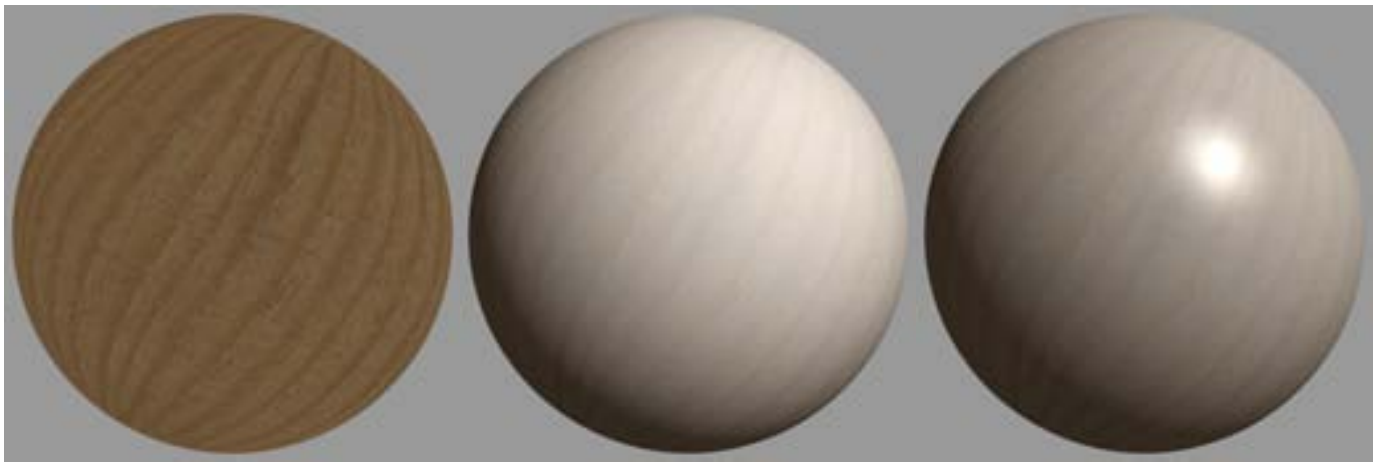
- Luz ambiente: luz que proporciona iluminación uniforme a lo largo de la escena.
- Reflexión difusa: luz reflejada por la superficie en todas las direcciones.
- Reflexión especular: luz reflejada por la superficie en una sola dirección o en un rango de ángulos muy cercano al ángulo de reflexión perfecta.

Luz Ambiente

Descripción

- La luz ambiente I_a que se observa en cualquier punto de una superficie es siempre la misma.
- Parte de la luz que llega a un objeto es absorbida por este, y parte es reflejada, y se modela con el coeficiente k_a, $0 \leq k_a \leq 1$.

$$I_a = k_a L_a \tag{1}$$

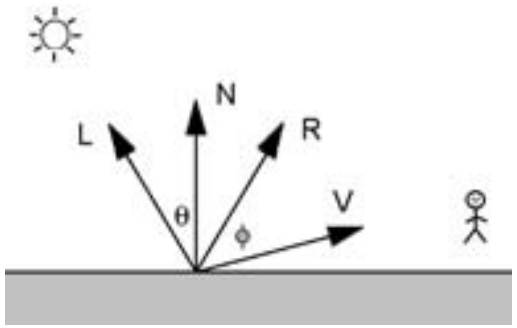


Reflexión Difusa

Descripción

- Es característica de superficies rugosas, mates, sin brillo.
- Se modela fácilmente con la Ley de Lambert. Así, el brillo observado en un punto depende sólo del ángulo θ, $0 \leq \theta \leq 90$, entre la dirección a la fuente de luz L y la normal N en dicho punto.
- L y N son vectores unitarios y k_d, $0 \leq k_d \leq 1$, representa la parte de luz difusa reflejada por la superficie.

$$I_d = k_d L_d \cos\theta = k_d L_d (L \cdot N) \tag{2}$$



Atenuación

Atenuación

- Al viajar desde su fuente de origen hasta el objeto situado a una distancia d.
- Los coeficientes a, b y c son constantes características de la fuente de luz.

$$I_d = \frac{k_d}{a + bd + cd^2} L_d (L \cdot N) \tag{3}$$

Reflexión Especular

Descripción

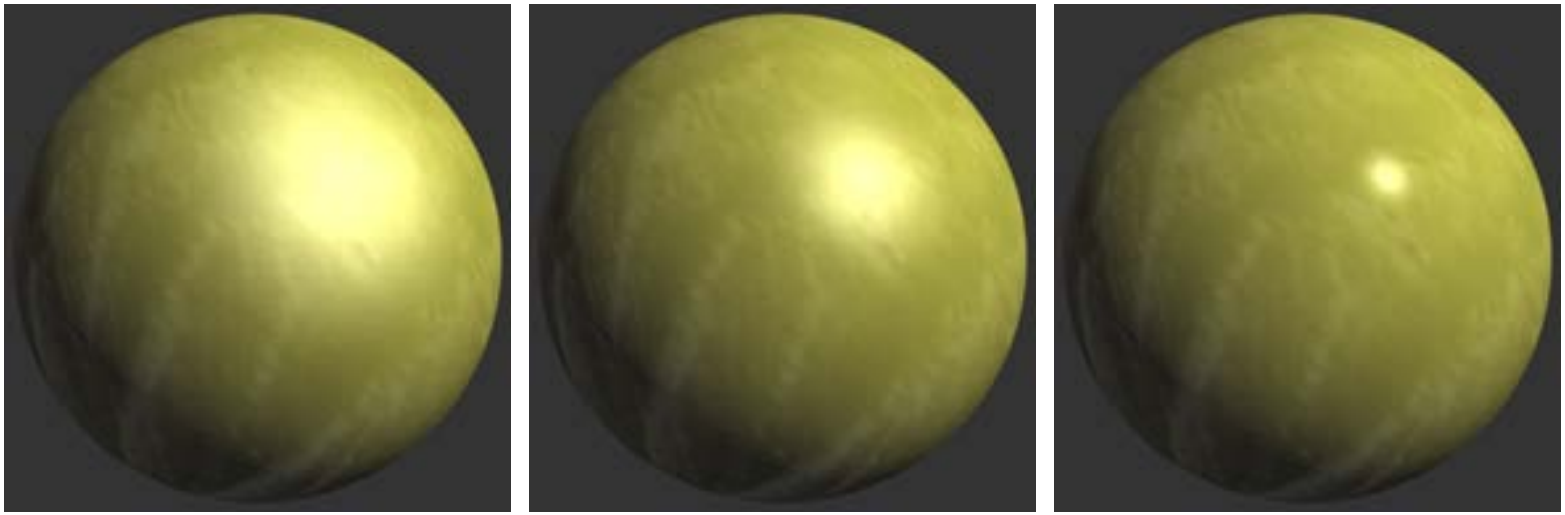
- Es propia de superficies brillantes, pulidas, y responsable de los brillos que suelen observarse.
- El color del brillo suele ser diferente del color de la superficie y muy parecido al color de la fuente de luz.
- Phong propone que la luz que llega al observador dependa únicamente del ángulo Φ entre el vector de reflexión perfecta R y el vector dirección del observador V.
- Si R y V son vectores unitarios, k_s, $0 \leq k_s \leq 1$, representa la parte de luz especular reflejada por la superficie y α modela el brillo característico del material de la superficie.

$$I_s = k_s L_s \cos^{\alpha} \Phi = k_s L_s (R \cdot V)^{\alpha} \quad (4)$$

$$R = 2N(N \cdot L) - L \quad (5)$$

El valor de α

- Un valor igual a 1 modela un brillo grande.
- Valores mucho mayores, por ejemplo entre 100 y 500, modelan brillos más pequeños propios de materiales, por ejemplo, metálicos.



(a) $\alpha = 3$ (b) $\alpha = 10$ (c) $\alpha = 100$

Materiales

En el modelo de Phong

- Se tiene en cuenta las propiedades del material del objeto al calcular la iluminación y así proporcionar mayor realismo.
- En concreto son cuatro:
  - ambiente k_a,
  - difusa k_d,
  - especular k_s
  - y brillo α.
- Por ejemplo:

Esmeralda: $k_a[] = \{ 0.022, 0.17, 0.02, 1.0 \}$
$k_d[] = \{ 0.08, 0.61, 0.08, 1.0 \}$
$k_s[] = \{ 0.63, 0.73, 0.63, 1.0 \}$
$\alpha[] = \{ 0.6 \}$

El modelo de Phong

En resumen

$$I = k_a L_a + \frac{1}{a + bd + cd^2}(k_d L_d (L \cdot N) + k_s L_s (R \cdot V)^{\alpha}) \qquad (6)$$

Listado 1: Función que implementa para una fuente de luz el modelo de iluminación de Phong sin incluir el factor de atenuación

```
// Ka, Kd, Ks, alfa, La, Ld y Ls son variables uniformes
// N, L y V se asumen normalizados
vec4 phong (vec3 N, vec3 L, vec3 V)
{
  float NdotL = dot(N,L);
  vec4  color = Ka * La;
  if (NdotL > 0.0)
  {
    vec3  R       = normalize(2 * N * NdotL - L);
    float RdotV_n = pow(max(0.0, dot(R,V)), alfa);

    color = color + ((NdotL * (Ld * Kd)) + (RdotV_n * (Ls * Ks)));
  }
  return color;
}
```

Tipos de Fuentes de Luz

Tipos

- Posicional: la fuente emite luz en todas las direcciones desde un punto dado, muy parecido a como por ejemplo ilumina una bombilla.
- Direccional: la fuente está ubicada en el infinito, todos los rayos de luz son paralelos y viajan en la misma dirección. En este caso el vector L en el modelo de iluminación de Phong es constante.

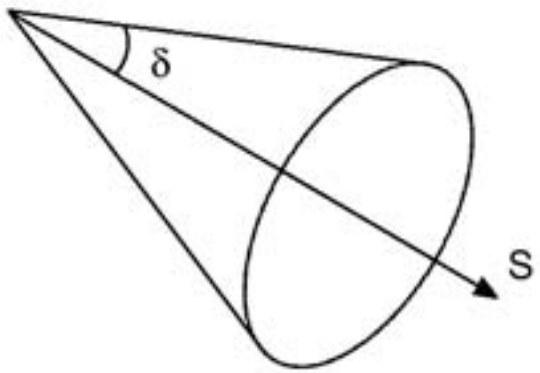
Foco de luz

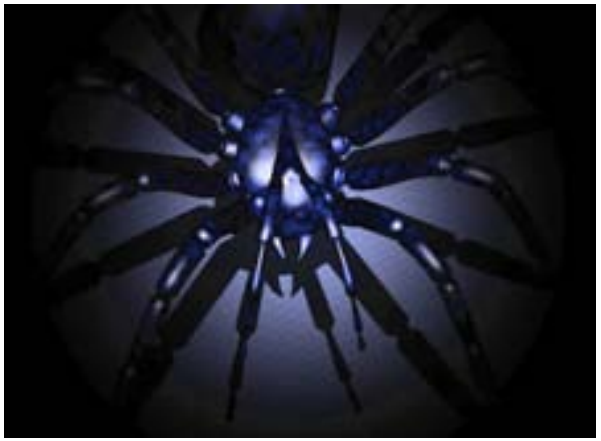
- Restringir los efectos de una fuente de luz posicional a un área limitada de la escena.

Foco de Luz

Cálculo

- Requiere una dirección S y un ángulo δ para la forma del cono.
- Un fragmento es iluminado si el ángulo entre el vector L y el vector S es menor que el ángulo δ.
- Atenuación: se calcula mediante el coseno del ángulo entre los vectores L y S elevado a un exponente. Cuanto mayor sea este exponente, mayor la concentración de luz alrededor del eje del cono.





Modelos de Sombreado

Métodos

- Plano: el modelo de iluminación se aplica una sola vez y su resultado se aplica a toda la superficie del polígono. Este método requiere la normal de cada polígono.
- Gouraud: el modelo de iluminación se aplica en cada vértice del polígono y los resultados se interpolan sobresu superficie. Este método requiere la normal en cada uno de los vértices del polígono.
- Phong: el modelo de iluminación se aplica para cada fragmento. Este método requiere la normal en el fragmento, que se puede obtener por interpolación de las normales de los vértices.

Ejemplos de sombreado

(d) Gouraud

(e) Phong

Shader para sombreado de Gouraud

```
// Vertex Shader ----------------------------------------------------
uniform mat4  projection, camera, transf; // matrices
uniform mat3  normal;
uniform vec4  Ka, Kd, Ks;                 // material
uniform float alfa;
uniform vec4  Lp, La, Ld, Ls;             // fuente de luz
in      vec3 posicion, vNormal;           // recibe vertice y normal
out     vec4 nuevoColor;                  // color del vertice

void main()
{
  vec4 ecPosition = camera*transf*vec4(posicion,1.0);
  vec3 N          = normalize(normal * vNormal);
  vec3 L          = normalize(vec3(Lp - ecPosition));
  vec3 V          = normalize(vec3(-ecPosition));

  nuevoColor  = phong(N, L, V);
  gl_Position = projection * ecPosition;
}

// Fragment Shader --------------------------------------------------
in  vec4 nuevoColor;
out vec4 colorFragmento;

void main()
{
  colorFragmento = nuevoColor;
}
```

Shader para sombreado de Phong

```
// Vertex Shader ------------------------------------------------------
uniform mat4 projection, camera, transf;
uniform mat3 normal;
uniform vec4 Lp;                    // fuente de luz
in      vec3 posicion, vNormal;     // recibe vertice y normal
out     vec3 N, L, V;               // los vectores

void main()
{
  vec4 ecPosition = camera*transf*vec4(posicion,1.0);

  N = normalize (normal * vNormal);
  L = vec3 (normalize(Lp - ecPosition));
  V = normalize (vec3(-ecPosition));

  gl_Position = projection * ecPosition;
}

// Fragment Shader ----------------------------------------------------
uniform vec4  Ka, Kd, Ks;           // material
uniform float alfa;
uniform vec4  La, Ld, Ls;           // fuente de luz
in      vec3  N, L, V;
out     vec4  colorFragmento;

void main() {
  colorFragmento = phong (N, L, V);
}
```