

Luz y sombreado

Tarea 04

Computación Gráfica

UNAM 2022-2

Gibran Zazueta Cruz
15/marzo/2022

1. Introducción

1.1. Modelo de luz

El modelo de luz utilizado se compone de una componente ambiental, una difusa y una especular.



Figura 1: Componentes de luz [1]

Para la implementación se utilizaron los siguientes parámetros

1.1.1. Ambiental

La luz ambiental es una luz que ilumina de manera homogénea todos los objetos de la escena. Se calcula como

$$I_A = \kappa_A \Lambda_A \quad (1)$$

En la implementación se utilizó $I_A = 100$ y $\kappa_A = [0,3, 0,3, 0,3]$

1.1.2. Difusa

La iluminación difusa le da mayor iluminación a las zonas del objeto alineadas hacia la fuente de luz. Es decir la luz muestra una especie de *dirección* sobre el objeto.

La luz difusa se calcula como

$$I_D = \kappa_D \Lambda_D (\vec{n} \cdot \vec{l}) \quad (2)$$

donde \vec{n} es la normal del vértice de incidencia y \vec{l} es el vector de dirección de la fuente de luz sobre el vértice.

En la implementación se utilizaron 2 fuentes de luz difusa. Una con luz blanca e intensidad $\Lambda_{D_1} = 10$. La otra con luz roja e intensidad $\Lambda_{D_2} = 80$. Ambas están localizadas en la posición $[10, 10, 10]$

El coeficiente κ del objeto se consideró como $\kappa_D = [0,07568, 0,61424, 0,07568]$

1.1.3. Especular

La luz especular está basada en la propiedad reflectiva de la luz al incidir sobre un objeto. Para la visualización de su efecto se deberá definir un observador dentro de la escena.

El cálculo es el siguiente

$$I_E = \kappa_E \Lambda_E (\vec{o} \cdot \vec{l})^\rho \quad (3)$$

donde \vec{o} es el vector de dirección del observador. La ecuación (3) también se puede escribir

$$I_E = \kappa_E \Lambda_E (\vec{o} \cdot \vec{h})^\rho \quad (4)$$

donde $h = 2(\vec{o} \cdot \vec{l})\vec{n} - \vec{l}$

Para la implementación se utilizó una luz especular blanca con intensidad 2 y en la posición $[10, 10, 10]$

El coeficiente del objeto para luz especular fue: $\kappa_E = 0,633, 0,72811, 0,66$;

1.2. Gouraud Shading

El sombreado de Gouraud es un método para sombrear la superficie de un polígono, de acuerdo a un modelo de luz implementado.

Los pasos son:

- Calcular los vectores normales en los vértices del polígono
- Calcular la iluminación en cada vértice
- Interpolarse el valor de iluminación entre vértices (formar border)
- Interpolarse la iluminación entre los bordes para rellenar el polígono (scan conversion)

1.3. Phong Shading

Phong es un método de sombreado más sofisticado que Gouraud.

La diferencia principal es que la interpolación entre vértices es del valor de sus normales

Los pasos para el sombreado de Phong son:

- Calcular los vectores normales en los vértices del polígono
- Interpolarse las normales sobre la superficie del polígono
- Calcular la iluminación en cada normal (punto) de la superficie.

Para esta implementación también se realizó interpolación lineal del vector \vec{l} y \vec{o}

2. Estructura del código

El programa tiene como base el código de la tarea 3.

En *lights.h* se define la clase *lights* para representar las luces de la escena. Cada objeto tiene intensidad, posición y color.

En *mainwindow.cpp* se crean los objetos de luz para luz blanca, luz roja y luz especular

En la clase *cubeobject* se calculan las normales de las caras del cubo. Después las normales de cada vertice, haciendo un promedio de las normales de sus caras colindantes.

En *CamProjection.cpp* la función *projectPoint* recibe como entrada el objeto (cuboide) y las luces de la escena. Además de proyectar los punto y realizar scan conversion, aqui se calcula la intensidad de la luz y el ocmbreado de la superficie.

Para gouraud se calcula el color en los 8 vértices del cuboide. Para ello se utiliza la función *calcLightVertex* de *shading.h* que implementa las ecuaciones (1), (2) y (3). Después se interpolan los valores de intensidad de los vértices en el mismo método de scan conversion

Para Phong, se calculan las normales \vec{n} de los vértices, el vector \vec{l} entre cada vértice con la posición de la luz y el vector \vec{o} entre cada vértice con la posición del observador. Se interpolan estos valores sobre la superficie dentro del método de scan conversion

3. Ejecutar el programa

En la carpeta de build se puede ejecutar el programa con el archivo *lightingShading-Run*. Desde la consola de comandos de linux:

```
bash
```

```
./lightingShading-Run
```

En la carpeta principal está el código fuente. Para generar el ejecutable primero se genera el Makefile con

```
bash
```

```
qmake lightingShading.pro
```

Después se construye el proyecto con *make*

4. Instrucciones de uso

Se presenta la interfaz del programa.

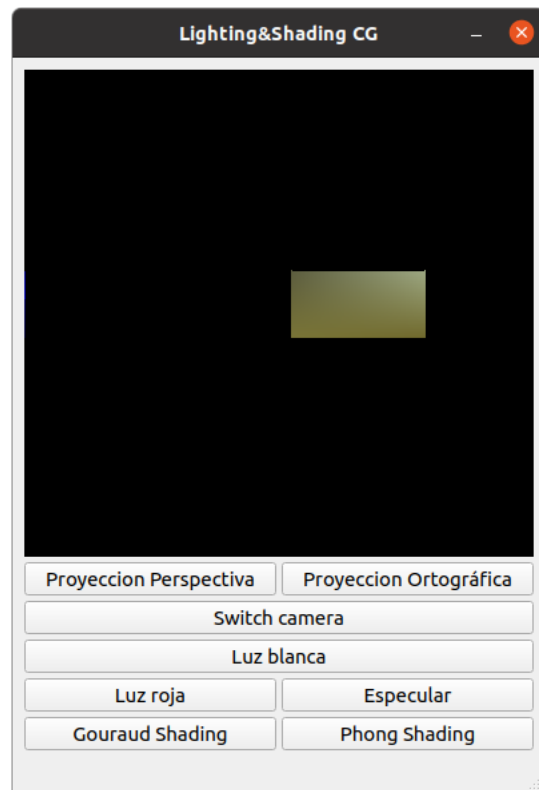


Figura 2: Interfaz gráfica del programa

Al presionar *Proyeccion Perspectiva* o *Proyeccion Ortografica* se cambia al tipo de proyección correspondiente.

SWitch camera cambia el renderizado entre la cámara 1 y la cámara 2

Los botones *Luz blanca*, *Luz Roja* y *Especular* encienden o apagan las luces correspondientes. El programa inicia con todas las luces activas.

Finalmente *Gouraud Shading* activa el combreado por Gouraud y *Phong Shading* el sombreado de Phong.

5. Programa en ejecución

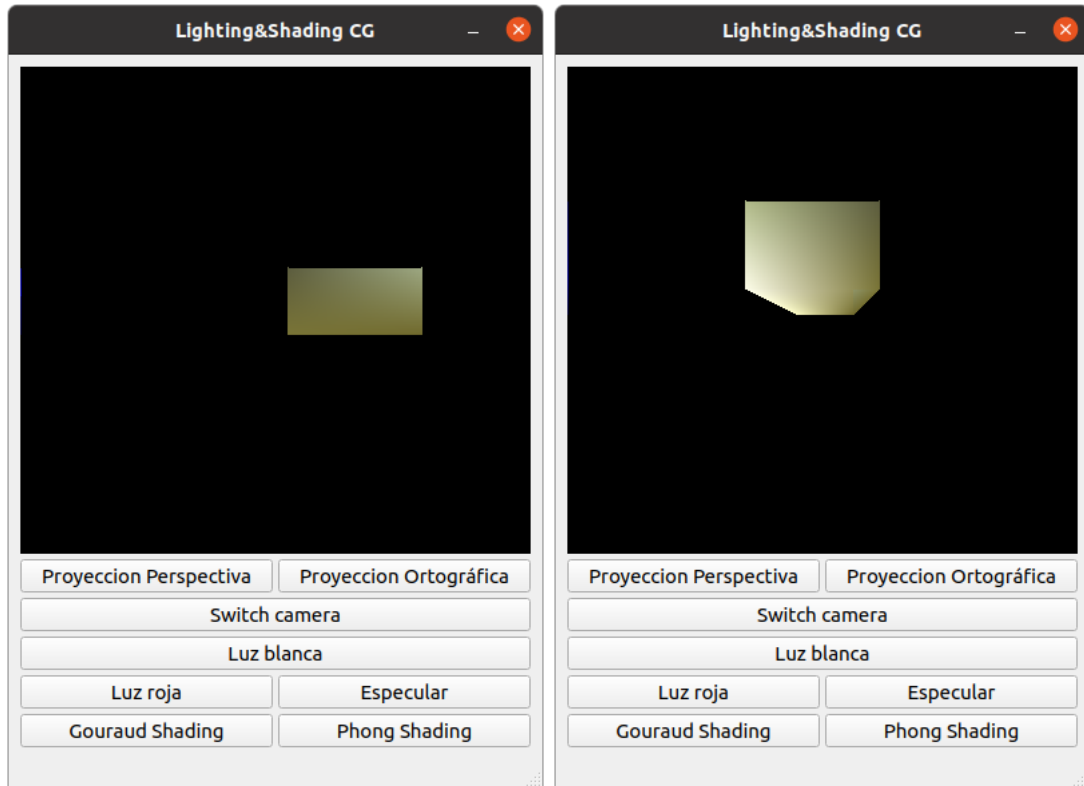


Figura 3: Gouraud Shading. Cámara 1 y 2. Proyectado en perspectiva

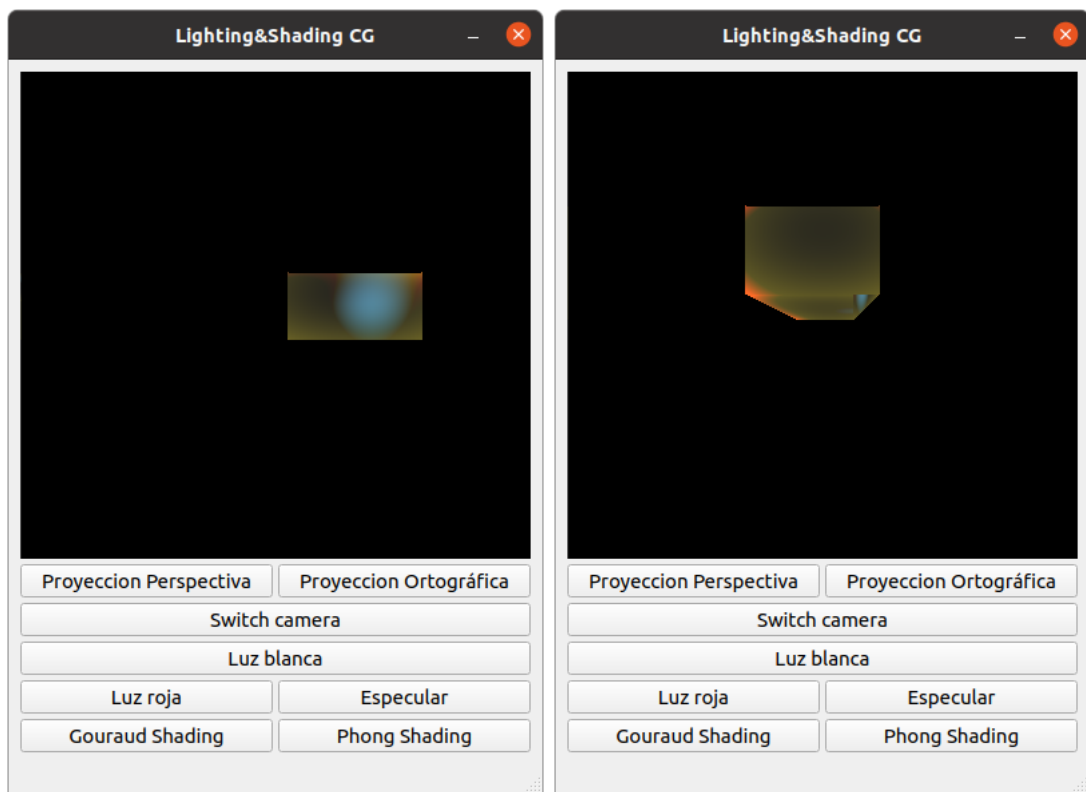


Figura 4: Phong Shading. Cámara 1 y 2. Proyectado en perspectiva

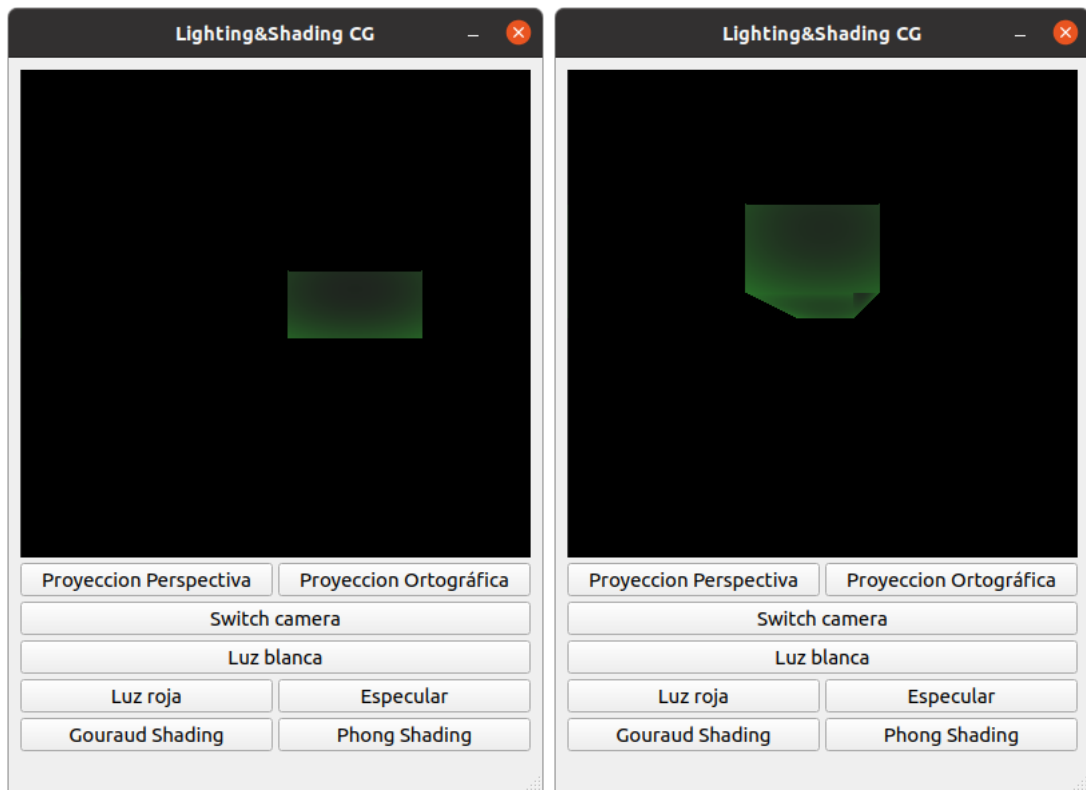


Figura 5: Phong Shading. Solo luz blanca. Cámara 1 y 2. Proyectado en perspectiva

Referencias

- [1] Learn OpenGL. Basic Lighting (*[https : //learnopengl.com/Lighting/Basic – Lighting](https://learnopengl.com/Lighting/Basic%20Lighting)*)