National University of Sciences and Technology (NUST)
School of Electrical Engineering and Computer Science

Department of Computing

CS361: Computer Graphics

Class: BSCS-2AB & BESE-3AB

Lab08: Shading

**Date: 16th November, 2015**

Time: 2:00pm- 5:00pm

Instructor: Dr. Muhammad Muddassir Malik

Lab09: Gouraud Shading

Introduction

In computer graphics, shading refers to the process of altering the color of an object/surface/polygon in the 3D scene, based on its angle to lights and its distance from lights to create a photorealistic effect. Shading is performed during the rendering process by a program called a shader.
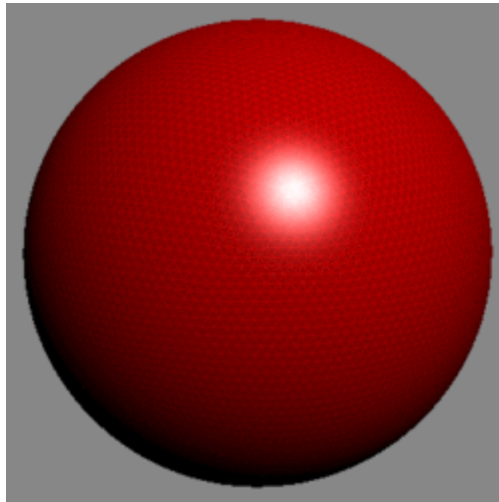
Objectives

The objective of this lab is to implement the Shading techniques.

Tools/Software Requirement

For testing HTML 5, CSS, JS

Description

Gouraud shading, named after Henri Gouraud, is an interpolation method used in computer graphics to produce continuous shading of surfaces represented by polygon meshes. In practice, Gouraud shading is most often used to achieve continuous lighting on triangle surfaces by computing the lighting at the corners of each triangle and linearly interpolating the resulting colours for each pixel covered by the triangle. Gouraud first published the technique in 1971

Gouraud shading works as follows

An estimate to the surface normal of each vertex in a polygonal 3D model is either specified for each vertex or found by averaging the surface normals of the polygons that meet at each vertex. Using these estimates, lighting computations based on a reflection model, e.g. the Phong reflection model, are then performed to produce colour intensities at the vertices. For eachscreen pixel that is covered by the polygonal mesh, colour intensities can then be interpolated from the colour values calculated at the vertices.
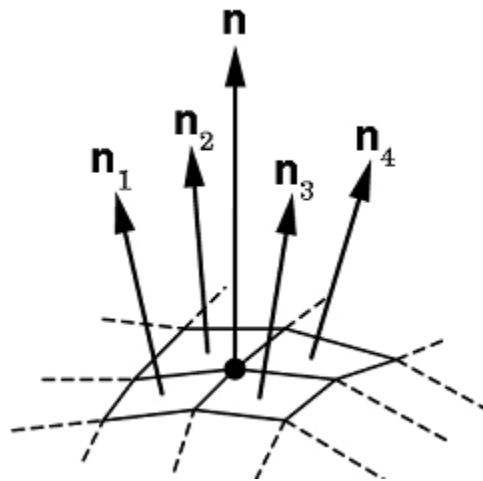
Gouraud shading

- Special case of interpolative shading
- How do we calculate vertex normals?
- Gouraud: average all adjacent face normal

$$n = \frac{n_1 + n_2 + n_3 + n_4}{|n_1 + n_2 + n_3 + n_4|}$$

Requires knowledge about which faces share a vertex—adjacency info

Task 1:
Implement Gouraud Shading
Create two cubes that can be rotated using keyboard. Implement Gauraud shading for the scene using one point light. Different sides of the cubes should have different material properties. **[7]**

Task 2: [2]
Implement Phong Illumination on the scene created for Task 1. Provide a toggle button to switch between Gauraud and Phong shadings using dat.Gui

Bonus Task: [1]
Implement two (point light and directional) lights in task 2 with options to vary all parameters of light, material and shading.

Deliverable
Upload your code with snap shots of the output.