Shading Models: Flat, Gouraud, Phong, and Simple Illumination Models

Real World Lighting Characteristics

● Light can be modeled as groups of traveling rays.
● When rays collide with a surface, some are absorbed and some are reflected. Black objects appear black because they absorb all incoming light--nothing is reflected. White objects appear white because all colors are reflected back and the combination of those colors produces white--nothing is absorbed. Most real world objects fall in the middle of this range.
● We perceive objects based on how rays of light strike the sensors in our eyes. The human eye can detect the wavelength (color) of the ray as well as where it originated from (position / angle). Combining this information gives us the sensation of vision. [1]

Lighting Models Used in Computer Graphics

● There are many approaches to simulating lighting in a computer. Some of these **lighting models** are more computationally expensive than others. Ray tracing, for instance, individually simulates light rays in a scene to render the lighting conditions. In general, however, this is too computationally expensive to be practical.
● Instead, various approximation techniques are made to capture the light in a virtual world. A common approach is to combine multiple lighting approximations into one final result. These include
  ○ Emissive Lighting
  ○ Ambient Lighting
  ○ Diffuse Reflection
  ○ Specular Reflection
[1]
● In OpenGL materials are described by specifying colors for each of the lighting components just mentioned along with transparency and a shininess factor.
  ○ Ambient Color (R, G, B)
    ■ The contribution of color due to ambient light.
  ○ Diffuse Color (R, G, B)
    ■ The contribution of color due to directed, point, and spotlights.
  ○ Specular Color (R, G, B)
    ■ The contribution of color due to specularity (usually white).
  ○ Emissive Color (R, G, B)
    ■ The contribution of color due to being self-lit.
  ○ Transparency
    ■ How see-through an object is.
  ○ Shininess
    ■ Controls the contribution of color due to the specular component.
[7]

- Even given these different lighting components, there is more than one way that we can apply them to scene geometry. Since virtual objects are made up of a series of polygons, we have to define appropriate ways to "paint" the light onto the objects so that the virtually lit world is realistic.
- The way that polygons are shaded depends on the **shading model**. We will give an overview of the differences between the Flat Shading, Gouraud Shading, and Phong Shading techniques.

Emissive Lighting

- When using emissive lighting, we do not require any lights to exist in a scene. Instead, each polygon is self-luminous. [1]
- Polygons light themselves, but do not cast light onto other objects. [1]
- It is not a very realistic approach, because the computed color is independent of the rest of the world; it is not influenced by other lights, other polygons, the position of the polygon in the scene, or any other factors. [1]
- Items do not usually appear to be 3D if neighboring faces are the same color because visually they all blend together into one shape. [1]
- Generally used as an additional effect to self-lit objects in a scene like a car headlight or a neon sign. [1]

Ambient Lighting

- Ambient lighting can be thought of as the base level of light in a room. [1]
- When you turn on an overhead light in a room, it lights up all (or most of) the room because the light rays do not propagate out in one direction and suddenly stop. Instead, most rays are reflected and travel off in a new direction and interact with a new surface and the process is repeated until all of the light energy has been absorbed. Because light bounces around in so many directions, most objects in the room will be lit to a similar level. The ambient lighting is this average.
- The light source is thought of as non-directional because we're considering the light that has already bounced around in all directions. [1]
- As with emissive lighting, the ambient light applied to an object is not affected by any other part of the scene, nor is it affected by the object's position or orientation. [1]
- Like with emissive lighting, ambient lighting does not produce a 3-dimensional effect on its own. [1]

Diffuse (Lambertian) Reflection

- Shiny objects like mirrors obviously have reflections--it's very easy to see because often there are bright highlights or you can see yourself in them. Dull objects, however, also reflect light--it just takes a different kind of appearance. This is what diffuse lighting models. [1]
- In general, a diffuse reflection describes how a single incoming ray of light becomes multiple outgoing rays of light reflected in multiple directions.

- The model used for diffuse lighting in computer graphics says that when light comes in from a specific direction, it is reflected with equal intensity in all directions. [1]
- The brightness of the reflection depends on angle between the direction the lit face is pointing and the direction the light is coming from. [1]
- The position of the viewer does **not** affect diffuse reflections. [1]
- The distance away from the light source may affect how the reflection is calculated if an attenuation factor is introduced. This is useful to model how light in atmosphere disperses over a distance--consider how a flashlight only allows you to see a finite distance into the woods. [1]

Specular Reflection

- Specular reflection is the counterpart to diffuse reflection, and is helpful for rendering shiny objects. [1]
- Things like chalk and paper have a very low specular component whereas metal and glass have a high specular component. [1]
- Think, for instance, of a balloon and how there is often one part of the balloon that appears shinier than the rest and is usually sketched as a little bright colored swatch. That is a specular highlight.
- Unlike a diffuse reflection which splits one ray of light into many, specular reflection assumes that the beam of light stays focused when it is reflected.
- The position of the viewer **is** important in rendering a specular highlight. [1]

Types of Light Sources Used in Computer Graphics

- For diffuse and specular lighting, we need to take into account the light sources present in the scene. [1]
- In computer graphics, the basic light options are generally point lights, directional lights, and spotlights. [1]
- Each light will be assigned a color and an intensity. [1]

Point Lights

- A point light is described by a single location in space. It gives of light equally in all directions. A real world approximation would be a bare lightbulb hanging from a string. [1]
- There is usually a falloff function that describes how the light loses its potency over a distance. [1]
- The angle at which a point like strikes and lights an object is a function of their relative positions. [1]

Directional Lights

- A directional light is a light source that sends out a stream of rays all in a single parallel direction. A real world approximation would be sunlight. [1]
- Can be thought of as a point light an infinite distance away, with no falloff function. [1]

● Does not lose intensity with distance. [1]

Spotlights

● A spotlight describes light that radiates from a location in space out in a cone shape. For a real world example, consider headlights on a car which cast light in a roughly similar manner. [1]
● Some spotlights have a falloff function that makes the light more intense at the center of the cone and softer towards the outside edges of the cone. [1]

Vector Normals - A Brief Overview

● A vector is a mathematical construct used to represent quantities with both magnitude and direction. [5]
● A normal vector is a vector that is perpendicular to a surface at any given point. [5]
● Normals are very important in computer graphics because they allow us to understand which way a facet of an object is facing and that, in turn, lets us compute how light strikes it and how it should be shaded.

Shading Models Used in Computer Graphics

● Even once we've defined built a workable lighting model, we have to apply light to objects inside of a scene. For applying light to all of the facets that make up an object in our virtual world, we must embrace a particular shading model. The three we will discuss are Flat, Gouraud, and Phong.

Flat Shading

● Flat shading (also known as faceted shading [6]) is, perhaps, the simplest shading model. With flat shading, each entire polygon is drawn with one color. [1]
● It tends to produce a blocky, unrealistic look.
● To calculate the color of the face, we need to know the normal of the polygon. Once per polygon the normal is used to calculate the influence of lights in the scene and then the entire facet is painted with the resulting color. [1]
● It is very inexpensive to compute. [1] It was used in early 3D video games like Star Fox and Virtua Racing because it was the only practical way to shade an object with the available computational resources at the time.
● It is frequently used for prototyping and testing because it can be rendered so efficiently.

Gouraud Shading

- Gouraud shading (also known as "color interpolation shading") is significantly more realistic than flat shading.
- While more expensive to compute than flat shading, Gouraud still comes at a relatively modest cost. [2]
- Instead of drawing each facet as a single color, it interpolates across the polygon when given the colors occurring at each point that defines the polygon. [1]
- To do the shading, we need to know the normal at each of the vertices of the polygon. Then, the lighting equation is computed at each vertex and the color of the vertex is computed. [1]
- Once we have the vertex colors, a bilinear interpolation (between vertices and along scanlines) is performed to figure out the color of any given point on the rendered polygon. This has to be down across each color channel. [2]
- **Computations per point plotted:** 3 floating point additions (for intensity interpolation)
- Gouraud shading does not handle specular highlights well and can sometimes look unnatural. [2]

Phong Shading

- Phong Shading (also known as "normal vector interpolation shading") is the most computationally expensive of the three tactics we will discuss [3], but it also produces the best results.
- Phong shading handles specular highlights much, much better than Gouraud. [1]
- Like Gouraud shading, Phong shading employs bilinear interpolation. But, instead of interpolating the colors at each vertex, it interpolates the normals as it paints each point on a polygon. [1]
- At each point on the facet being drawn, it takes the interpolated normal and computes the light cast by the light sources in the scene. [3] It should be clear that this is much more costly.
- Phong shading combines diffuse, ambient, and specular lighting for each point that it draws. [4]
- **Computations per point plotted:** 3 float point additions (normals) + (3 squares + sqrt for normal) + (recompute intensity - 3 * (~10-20 floating point multiplications depending on degree of approximation)). Much more than Gouraud, which only has 3 floating point additions per pixel. [3]

References

[1]
University of Illinois at Chicago course notes page on Lighting and Shading.
Gave details on simple illumination models and some info on Flat, Gouraud, and Phong.
source

https://www.siggraph.org/education/materials/HyperGraph/scanline/shade_models/shading.htm
Siggraph shading models index page.
>   From it:
>   [2]
>   Siggraph Gouraud page for Scan-Line graphics.
>   Lists the algorithm, disadvantages, advantages, and visual representations.
>   source
>   [3]
>   Siggraph Phong page for Scan-Line graphics.
>   Lists the algorithm and visual representations.
>   source

[4]
Siggraph Phong model for Specular Reflection page.
Gave computational information about Phong shading and specular illumination.
source

[5]
Excerpt of Algorithms of Informatics.
Gave info on general computer rendering and shading
source

[6]
Pace University lecture page.
Detailed computational and algorithmic descriptions of simple illuminations (Ambient, Diffuse,
        and Specular), Gouraud, and Phong shading.
source

[7]
Ohio State University Lecture slides.
Gave shading model information in regards to the OpenGL library.
source