Lighting in OpenGL
CSC 461: Introduction to Computer Graphics

1 Lighting

Lighting determines the colours of the vertices in the scene based on light properties and materials properties assigned to difference surfaces in the scene.

There are four steps involved in using lighting in OpenGL:

1. Define normals for every vertex of each object.
2. Create and position one or more light sources.
3. Create and initialize the lighting model.
4. Define materials properties for objects in the scene.

In order for diffuse and specular lighting calculations to work properly, normals must be specified for each vertex in the scene with:

```c
glNormal3f( x, y, z )
```

where \((x, y, z)\) is the direction of the normal.

Normals to a polygon can be computed as the cross-product of any two adjacent, non-colinear edges of the polygon. Normals are specified prior to the vertex they correspond to:

```c
glNormal3f( nx, ny, nz );
glVertex3f( vx, vy, vz );
```

1.1 Light Sources

A light source is defined by its position, and by the intensity of the ambient, diffuse, and specular light that it emits into the scene. All light properties are setup with:

```c
glLightfv( light, param, value )
```

where `light` defines which light’s properties are being set (this runs from `GL_LIGHT0` through `GL_LIGHT7`), `param` specifies which parameter of the light is being set, and `value` holds the value(s) of the parameter. Common parameters include:

- `GL_AMBIENT`: \((r, g, b, a)\) intensity of ambient contribution, default is \((0, 0, 0, 0)\),
- `GL_DIFFUSE`: \((r, g, b, a)\) intensity of diffuse contribution, default is \((1, 1, 1, 1)\),
- `GL_SPECULAR`: \((r, g, b, a)\) intensity of specular contribution, default is \((1, 1, 1, 1)\), and
- `GL_POSITION`: \((x, y, z, w)\) position of light, default is \((0, 0, 1, 1)\).
Note that for GL\_POSITION, if $w = 0$ then the light source is assumed to be at infinity. This simplifies lighting calculations since the direction from any vertex in the scene to an infinite light source is constant. If $w = 0$, light source is \textit{“local”} to the scene, and the direction to the light source must be computed at each vertex.

Also, light source position is treated as a geometric primitive—a vertex—so it is subjected to the same matrix transformations as any other primitive. This means you can manipulate the light’s position via the MODELVIEW stack just as you would for any other vertex.

Once a light source’s properties are set, you must \textit{“turn on”} the light with:

\begin{verbatim}
    glEnable( GL\_LIGHT0 )
\end{verbatim}

### 1.2 Lighting Model

The lighting model allows you to set the following properties of the scene:

- a global ambient light intensity with \texttt{glLightModelfv(GL\_LIGHT\_MODEL\_AMBIENT, amb)} (default is $(.2, .2, .2, 1)$), where \texttt{amb} is a $(r, g, b, a)$ float array defining the ambient intensity,
- whether the viewpoint is local or infinite (default is infinite to provide a constant direction from any vertex to the viewer), using \texttt{glLightModeli(GL\_LIGHT\_MODEL\_LOCAL\_VIEWER, GL\_TRUE)}, and
- whether only front or both front and back faces of polygons should have lighting performed (default is front faces only), using \texttt{glLightModeli(GL\_LIGHT\_MODEL\_TWO\_SIDE, GL\_TRUE)}.

Once the lighting model is set, lighting must be \textit{“turned on”} with:

\begin{verbatim}
    glEnable( GL\_LIGHTING )
\end{verbatim}

### 1.3 Materials Properties

In lighting mode, colours for vertices are not specified directly. Rather, materials properties are defined for a surface or vertex, and those are combined with the vertex normal, direction to the light source, direction to the viewer, and light properties to determine the colour of the surface.

Materials properties are set with:

\begin{verbatim}
    glMaterialfv( face, param, value )
\end{verbatim}

where \texttt{face} defines the face to set (GL\_FRONT, GL\_BACK, or GL\_FRONT\_AND\_BACK), \texttt{param} specifies which parameter of the surface to set, and \texttt{value} holds the value(s) of the parameter.

Common parameters include:

- \texttt{GL\_AMBIENT}: $(r, g, b, a)$ of ambient reflection, default is $(.2, .2, .2, 1)$,
- \texttt{GL\_DIFFUSE}: $(r, g, b, a)$ of diffuse reflection, default is $(.8, .8, .8, 1)$,
- \texttt{GL\_SPECULAR}: $(r, g, b, a)$ of specular reflection, default is $(0, 0, 0, 1)$, and
- \texttt{GL\_SHININESS}: specular exponent, default is 0.

As with \texttt{glColor}, \texttt{glMaterial} is used to set materials properties for all vertices falling the call, up to the next \texttt{glMaterial} call.