Lecture 14: Shadows & Interpolation Shading

Adam Hawley

December 22, 2018

Contents

1 Shadowing

There are two ways a point can be in a shadow: firstly, points can be in shadow because they face away from the light source or they can be in darkness caused by another point. The first is called an self/attached shadow while the second is a cast shadow.

1.1 Self-Shadowing

How can we test for self shadows? The check is based on whether the angle between the surface normal and the light source is greater than 90°:

$$\theta_i > 90^\circ \implies \text{Self shadow}$$

Equivalently, check whether the dot product is negative:

$$\mathbf{n} \cdot \mathbf{s} < 0 \implies \text{Self shadow}$$

This is a very efficient process as it depends on local information only. Normals are already computed for the reflectance calcultion, simply check dot product.

Self shadowing can be incorporated into the Lambertian model since Lambertian shading uses the same calculation. (The dot product between the surface normal and the light source) To make use of the pre-calculated dot product, simply clamp negative values to zero:

$$I = max(0, \mathbf{n} \cdot \mathbf{s})$$

Note: we did not consider this in either the shape-from-shading or photometric stereo algorithms in the previous lecture. So self shadows are not accounted for. "Add noise to the solution"

1.2 Cast Shadows

Case shadows are much more difficult to calculate. A point can be shadowed by a distant part of the surface. (They depend on global geometry) To check whether a point is in shadow, we must consider all other points on the surface.

If we set height values for every pixel (x, y) as $z(x, y)$. Then the cast shadow test can be written as:

$$\forall d \in \mathbb{R}^+, z(x, y) + ds_z > z(x + ds_x, y + ds_y) \implies \text{Not in cast shadow}$$

For general polygonal surfaces it is more complex: For every polygon or vertex, compute equation of line to light source. Then test whether the line intersects any other polygon in the scene e.g:

- Compute plane equation for every polygon
- Compute line/plane intersection
- Test whether intersection is inside or outside extent of polygon

There are efficient ways to calculate this.

2 Flat Shading

For each planar polygon we compute the surface normal. The surface normal is constant over the polygon and used to compute a single colour for the polygon. The whole polygon is then filled with this colour. For objects which consist of only planar surfaces, it can look fine. However it generally looks bad for polygonal approximations to curved surfaces.

Flat shading has a very bad effect with curved surfaces. **Mach banding** caused by **lateral inhibiion** means that the human eye effectively has a built in edge detector. Perception of change in intensity exaggerates actual change.

To smoothly vary the colour of a polygon across the surface we compute a surface normal for each vertex. The meaning of the normal direction to a point is not well defined so there are two options:

- If the surface is described by a parametric function (e.g a sphere or cylinder), just **derive an analytical expression for the surface normal** from the function.
- If the surface is an arbitrary polygonal mesh, compute **one normal per polygon and combine** these in some way to find vertex normals.

3 Face Normals

Typically, a surface is represented by a list of vertices and a list of faces. Faces are specified such that vertex ordering is clockwise or counterclockwise when viewed from outward facing side. To compute a normal to a face, use the cross product of two edge vectors:

$$\mathbf{n} = \mathbf{e}_1 \times \mathbf{e}_2$$

Remember: The length of $\mathbf{n}$ gives twice the area of the triangle. To find $\mathbf{n}$ as a unit vector simply divide it by its magnitude.

4 Vertex Normals

A vertex normal is a combination of the surface normals of the facets adjacent to the vertex.

To find a vertex normal, one can find the average of the surrounding facet normals. Often a weighted average is used — weights are areas of facets. Normals of larger facets contribute more to the vertex normal. If we use the unnormalised cross products of facet edges, we get this weighting for free. This method is fine for most purposes but there are more complex methods justified by differential geometry.

5 Interpolation Shading

There are two main approaches to use vertex normals for interpolation shading:

- **Gouraud**: calculate a colour for each vertex, **interpolate colours** along the polygon edge, then interpolate along scanlines within polygons.
- **Phong**: interpolate the vertex normals along the polygon edge, then **interpolate normals**

5.1 Drawbacks of Interpolation Shading

There are two main drawbacks of interpolation shading:

- The polygonss are still flat and so the silhouette of curved objects can visibly consist of straight lines.
- Shading is carried out after the projection transformation is applied. So if a perspective transformation is used the non-linear transformation of the Z axis may produce odd effects.

Both of these effects are reduced by using smaller polygons (i.e a higher resolution mesh).

Other possible problems include strange effects if the normals are set up wrongly. Mach banding may still be visible — although colours along polygon edge now match, the rate of change of colour may change drastically giving the appearance of an edge.

5.2 Gouraud Shading

Summary:

1. Each vertex has a surface normal and material properties, pass the reflectance model to compute a colour.
2. The colouring along the edge of each polygon is interpolated along the edge.
3. When the polygon is scan converted, interpolate between colours at edge along scanline.

5.2.1 Drawbacks of Gouraud Shading

The weakness of Gouraud shading is that it assumes the intensity changes smoothly within a polygon. Specular reflections cause large changes in intensity over a small distance. If the size of the specularity is small compared to the size of the polygon, Gouraud shading will miss it — this results in strange behaviour.

5.3 Phong Shading

In Phong shading we **interpolate surface normals**. The reflectance model will be evaluated independently for each pixel.

1. Interpolate normals along the polygon edges
2. Interpolate normals along scanlines, giving a normal for each pixel.
3. Calculate the colour intensity for each pixel.

5.3.1 Advantages & Disadvantages of Phong Shading

Advantages:

- Correct specular highlight
- Reduced Mach banding problems

Disadvantages:

- Expensive to compute

In modern GPUs something very similar to this happens in pixel or fragment shaders.

6 Conclusions

What you should know:

- Shadows: Cast vs. Self
- Face/Vertex normals and how to calculate them
- Purpose of interpolation shading
- Gouraud vs. Phong Shading