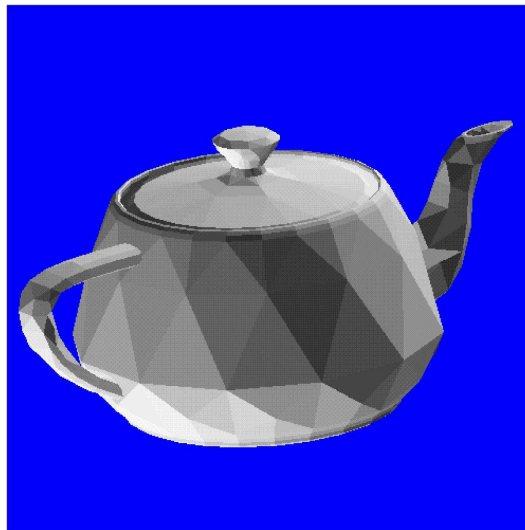


## Shading

Shading algorithms apply lighting models to polygons, through interpolation from the vertices.

OPENGL:

```
glShadeModel(GL_FLAT)
```



*Gouraud Shading:* Lighting is only computed at the vertices, via reflection illumination model, and the RGB reflection intensities are interpolated across the (convex) polygon

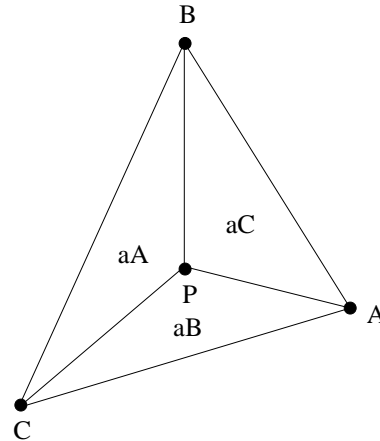
*Phong Shading:* A normal is specified at each vertex, and this normal is interpolated across the polygon. At each pixel, a lighting calculation is performed based on reflection illumination model.



## Gouraud Shading

- *Gouraud shading* interpolates colors across a polygon from the vertices
- Lighting calculations are only performed at the vertices
- Highlights can be missed or blurred
- Common in hardware renderers; model that OpenGL supports
- Gouraud shading is well-defined only for triangles...  
Equivalent to a *barycentric combination*
- Barycentric combinations are also *affine combinations*...  
Triangular Gouraud shading is *invariant* under affine transformations

## Triangle Gouraud Shading



$$\alpha_A = D(P, B, C) / D(A, B, C)$$

$$\alpha_B = D(A, P, C) / D(A, B, C)$$

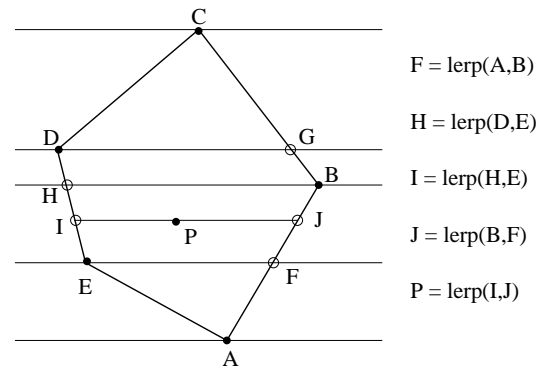
$$\alpha_C = D(A, B, P) / D(A, B, C)$$

$$\alpha_A + \alpha_B + \alpha_C = 1$$

$$P = \alpha_A A + \alpha_B B + \alpha_C C$$

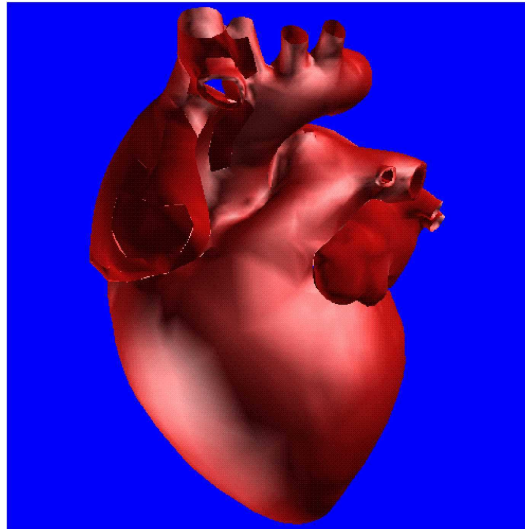
$$D(A, B, C) = \begin{vmatrix} 1 & 1 & 1 & 1 \\ 1 & x_A & y_A & z_A \\ 1 & x_B & y_B & z_B \\ 1 & x_C & y_C & z_C \end{vmatrix}$$

- Gouraud shading for polygons with more than three vertices:
  - Sort the vertices by  $y$  coordinate
  - Slice the polygon into trapezoids with parallel top and bottom
  - Interpolate colors along each edge of the trapezoid...
  - Interpolate colors along each scanline



- Gouraud shading gives *bilinear* interpolation within each trapezoid
- Since rotating the polygon can result in a different trapezoidal decomposition, *n*-sided Gouraud interpolation is *not affine invariant*

glShadeModel(GL-SMOOTH)

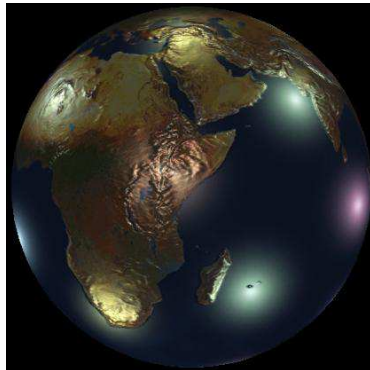


## Phong Shading

- *Phong Shading* interpolates lighting model parameters, *not* colors
- Much better rendition of highlights
- A *normal* is specified at each vertex of a polygon
- Vertex normals are independent of the polygon normal
- Vertex normals should relate to the surface being approximated by the polygon
- The normal is interpolated across the polygon (using Gouraud techniques).
- At each pixel,
  - Interpolate the normal...
  - Interpolate other shading parameters...
  - Compute the view and light vectors...
  - Evaluate the lighting model
- The lighting model does not have to be the Phong lighting model...
- Normal interpolation is nominally done by vector addition and renormalization
- Several “fast” approximations are possible



- The view and light vectors may also be interpolated or approximated



## Next Time: Global Illumination



## Reading Assignment and News

Chapter 6 pages 304 - 320, of Recommended Text.

Please also track the News section of the Course Web Pages for the most recent Announcements related to this course.

(<http://www.cs.utexas.edu/users/bajaj/graphics25/cs354/>)