Rendering Surfaces - Realism

- Model-view and projection transformations
- Visible surface determination
  - Compute set of surfaces visible from the viewpoint
- Illumination and shading (local, direct illumination-models): Render depth, lighting effects, material properties to improve 3D perception.
  - Know or can compute: view direction (V), light direction (L), local surface geometry (hence a normal N at a point p).
  - Desidered surfca reflectance R at a point p, in general, is a function of N,V,L and material properties.
  - Shading computation: get intensity or color at some projected object surface point, given illumination and shading models
- Global illumination models, radiosity
- Highlights
- Texture and bump (wrinkle) mapping
- Shadows
- Reflections and refractions
- Transparency and translucency
- Modeling curved surfaces, physics-based models, fractals

Review

Last lecture focused on direct illumination models.

- Light goes directly from the source to the object.
- No shadows, no reflections from other objects.

Illumination models express how light affects a surface's apparent color at a given point.

- *Ambient* term: uniform indirect illumination.
- *Diffuse* term: reflection that is the same in all directions, as if from a dull surface.
- *Specular* term: reflection that has a directional highlight, as if from a shiny surface.
- Summing these terms approximates reflection that is between the extremes of diffuse and specular.
- Phong model
  - point light source
  - only light intensity taken into account
  - all surface geometry but normal at point is ignored
  - difused and specular terms modeled locally
  - empirical model for specular term around direction of perfect reflection
  - in original Phong model, only white highlights
  - global (ambient) term is constant
  - plastic like, "floating" objects, no shadows
  - eroneous highlights for concave objects

Shading Models

Shading models use illumination models to determine the color with which the surface will be rendered.

Ideally, the renderer should apply the illumination model at every visible point on each surface.

This approach requires too much computation.

As an alternative, use sampling.

- Apply the illumination model at a subset of points.
- Interpolate the intensity at projections of the rest.

Make difference between illumination model (for calculating color at a point on the surface), and using such model embeded in algorithm for calculating color at projected pixels.

The Phong illumination model most commonly applied to polygonal mesh modes for creating of "realistic" images.

Incremental techniques:

- Gourad(1971)
- Phong(1975)
- combination of both for speed up of Phong

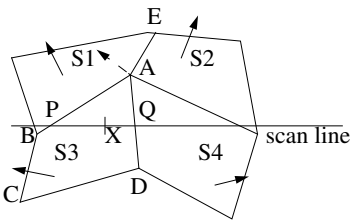
CG low: computation grows exp with appearance

Overview

1. *Flat* shading: apply an illumination model once per polygon.
2. *Gouraud* shading: apply an illumination model at each polygon vertex, interpolating the intensity in between.
   - approximates light reflections from curved surfaces
   - usually restricted to diffuse component
   - even if specular component considered, completely misses highlights in polygon interiors
   - simple, but does not elliminate poligone appearance
3. *Phong* shading: apply an illumination model at each pixel, interpolating the normal vector at points projected in pixels from normals at vertices.
4. Read: HB Ch 14-5

Gouraud Shading

- Goal: compute shade (color) at X.
- Solution: interpolate shade at X from shades computed at vertices based on some illumination model.



- Shades for individual polygones are matched with shades of neighboring polygons along common edges and vertices.
- Gourad shading algorithm:

1. For each vertex find avg unit normal (over all polygons that share the vertex) in world coordinates. Approximates true normal at the vertex.
2. Based on calculated normals apply illumination model and calculate shade at vertices
3. Linearly interplate to obtain the shades for non-vertex points

Recalculate after clipping (1 and 2).

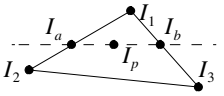
Gouraud shading

- The interpolation stage

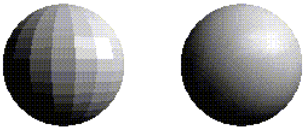
1. Vetex-vertex interpolation over edges
   - shade(Q)=weighted avg shade(A,D)
   - shade(P)=weighted avg shade(A,B)
2. Scan-line interpolation for interior points
   - shade(X) = weighted avg shade(P,Q)

Interpolate intensities as part of scan conversion.

- Interpolate span endpoint color from edge vertices.
- Interpolate pixel color within a span from span endpoints.
- Perform interpolation through incremental updates.



Some resulting images:



fl at Gouraud

- Gouraud shading in OpenGL:
  `glShadeModelModel(GL_SMOOTH);`

Gouraud shading

- reduces Mach bands (but not entirely).
- misses interior highlights
- smears highlights along edges
- some repetative 3D paterns can be missed completly

Flat (constant) shading

Bouknight (1970)

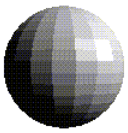
Sample illumination at one point per polygon.

Use constant interpolation: all other points on the polygon get that point's intensity.

This approach would be valid if:

- The true surface really is faceted, so **N** is constant.
- The light source is at infinity, so **L** is constant.
- The viewer is at infinity, so **V** is constant.

A resulting image:



- Flat shading in OpenGL, just enable it:

```
glShadeModel(GL_FLAT);
```


Phong shading

- Interpolate *normals* , not *shades,* then compute shade by some method.
- Apply the illumination model at each pointed projected.
- Normals' interpolation tends to " recover" curvature
- Normal interpolation in world coordiantes

Interpolate normal, **N**, for point projected at pixel as part of scan conversion.

- Interpolate normals at vertices as for Gouraud
- To get **N** at span endpoints, interpolate from edge vertices' normals.
- To get **N** within a span, interpolate from span endpoints.
- Perform interpolation through incremental updates.

Normal vector interpolation can cause problems

- Non-convex faces may lead to problems.
- Interpolation may mask regular changes

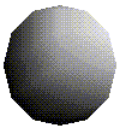
Phong shading avoids some errors in specular illumination that appear with Gouraud shading.

- Gouraud misses specular highlights within polygons.
- Gouraud spreads specular highlights along edges.

OpenGL does not support Phong shading.

Problems common to Gouraud and Phong shading

Silhouette edges are not smoothed.



Normal vector interpolation can cause problems:

Interpolation may mask regular changes

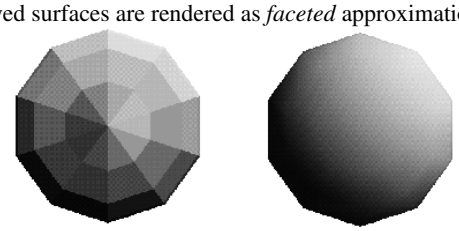
Interpolation ignores perspective distortion.

- Due to foreshortening, a change in scanlines does not correspond to constant change in z in OpenGL eye coordinates.
- So the scanline halfway between two vertices does *not* correspond to z halfway between the vertices' zs.
- But pixels on that scanline get an interpolated quantity (intensity or **N**) that *does* correspond to the halfway z.

For both methods: must be careful which polygon faces are used in shade or normal intrpolation (edges are softened, but in some cases we want to avoid this, i.e. edge between cylinder base and wall).

Vertex normal vectors

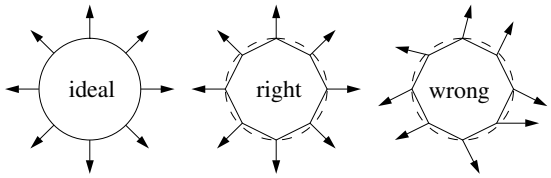
Curved surfaces are rendered as *faceted* approximations.



OpenGL supports smooth shading of faceted objects.

Smoother shading uses *normal vector*s.

- At a point on the ideal surface, the normal vector points straight out.



An OpenGL vertex can have a *vertex normal vector.*

- It should match the ideal surface's normal vector at that location.
- OpenGL smoothes the shading at the vertex, as if it were shading the ideal surface.

Vertex normal vectors

Calling `glNormal3f(`*`nx, ny, nz`*`)` sets the *current* vertex normal.

- It applies to subsequent vertices until it is changed.
- (*nx, ny, nz*) should have unit length.
- transformations may change normals lenght

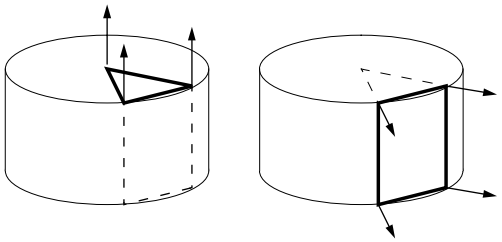
Vertex normals at creases

Crease edges should *not* have smooth shading.

Use *multiple* vertices, each with a *different* vertex normal.

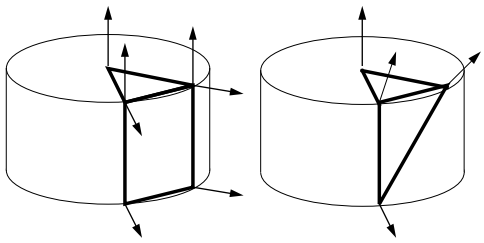
Example--cylinder in pieces:

- The top is one piece (one set of triangles).
- The sides are another piece (one set of quadrilaterals).



Vertex normals at creases

Thus, sharing vertices on creases does *not* work.



right wrong (triangle strip)