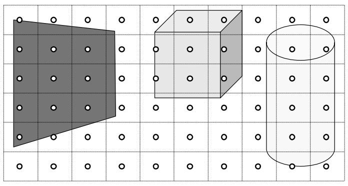
Ray Casting

- Simplest shading approach is to perform independent lighting calculation for every pixel



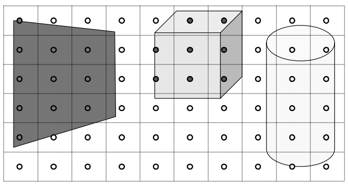
$$I = I_E + K_A I_{AL} + \sum_i (K_D (N \bullet L_i) I_i + K_S (V \bullet R_i)^n I_i)$$

Polygon Rendering Methods

- Given a freeform surface, one usually approximates the surface as a polyhedra.
- How do we calculate in practice the illumination at each point on the surface?
- Applying the illumination model at each surface point is computationally expensive.

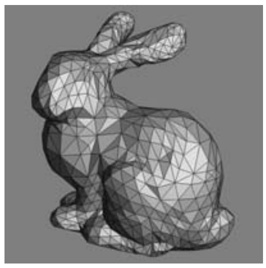
Polygon Shading

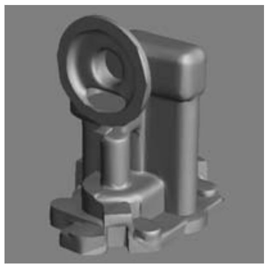
- Can take advantage of spatial coherence
  - Illumination calculations for pixels covered by same primitive are related to each other



$$I = I_E + K_A I_{AL} + \sum_i (K_D (N \bullet L_i) I_i + K_S (V \bullet R_i)^n I_i)$$

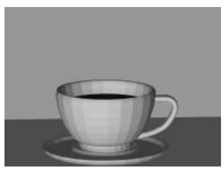
Piecewise linear approximation





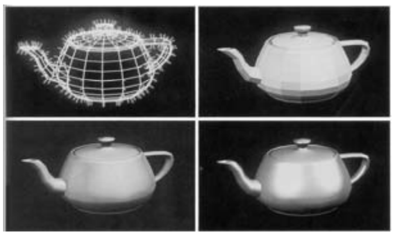
Polygonal Approximation

Smooth Shading



Polygon Shading Algorithms

Wireframe Flat

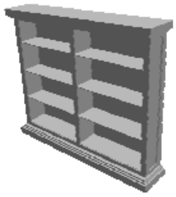


Gouraud Phong

Watt Plate 7

Flat Shading

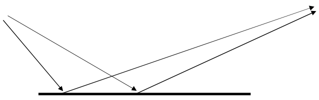
What if a faceted object is illuminated only by directional light sources and is either diffuse or viewed from infinitely far away



One illumination calculation per polygon
Assign all pixels inside each polygon the same color

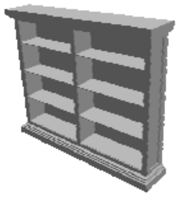
Flat Shading

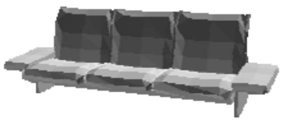
- A fast and simple method.
- Gives reasonable result only if all of the following assumptions are valid:
  - The object is really a polyhedron.
  - Light source is far away from the surface so that N•L is constant over each polygon.
  - Viewing position is far away from the surface so that V•R is constant over each polygon.



Flat Shading

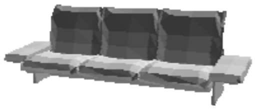
Objects look like they are composed of polygons
OK for polyhedral objects
Not so good for ones with smooth surfaces



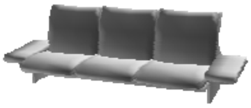


Gouraud Shading

- Produces smoothly shaded polygonal mesh
  - Piecewise linear approximation
  - Need fine mesh to capture subtle lighting effects

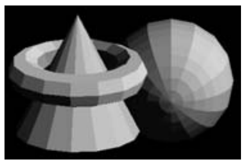


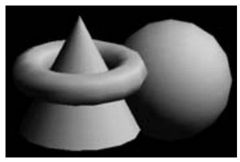
Flat Shading



Gouraud Shading

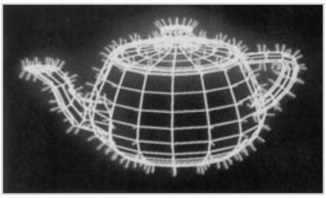
Polygon Smooth Shading





Gouraud Shading

- What if smooth surface is represented by polygonal mesh with a normal at each vertex?

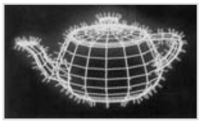


Watt Plate 7

$$I = I_E + K_A I_{AL} + \sum_i (K_D (N \bullet L_i) I_i + K_S (V \bullet R_i)^n I_i)$$

Gouraud Shading

- Smooth shading over adjacent polygons
  - Curved surfaces
- Renders the polygon surface by linearly interpolating intensity values across the surface.

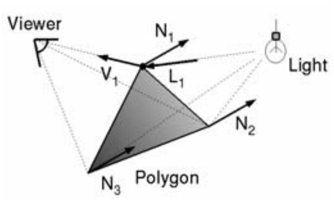


Mesh with shared normals at vertices

Watt Plate 7

Gouraud Shading

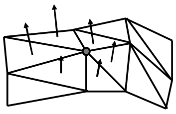
- One lighting calculation per vertex
  - Assign pixels inside polygon by interpolating colors computed at vertices



Gouraud Shading

1. Determine the average unit normal at each polygon vertex.
2. Apply an illumination model to each vertex to calculate the vertex intensity.
3. Linearly interpolate the vertex intensities over the surface polygon.

The normal vector at a vertex



The normal N_v of a vertex is an average of all neighboring normals:

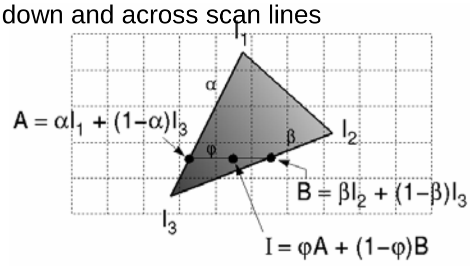
$$N_v = \frac{\sum_k N_k}{|\sum_k N_k|}$$

Which is simply the following normalized vector:

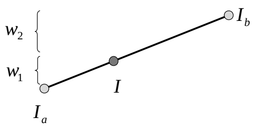
$$N_v = \sum_k N_k$$

Bilinear Interpolation

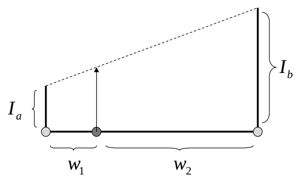
- Bilinearly interpolate colors at vertices down and across scan lines



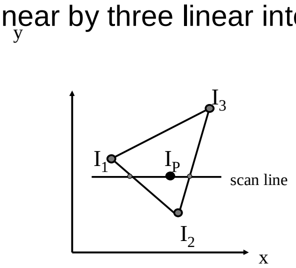
Linear Interpolation



$$I = w_2 I_a + w_1 I_b$$
$$I = w_2 I_a + (1 - w_2) I_b$$
$$I = w_2 (I_a - I_b) + I_b$$



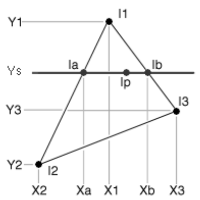
Bilinear by three linear interpolations



Two linear interpolations along the y-axis, and one along the x-axis.

Bilinear Interpolation

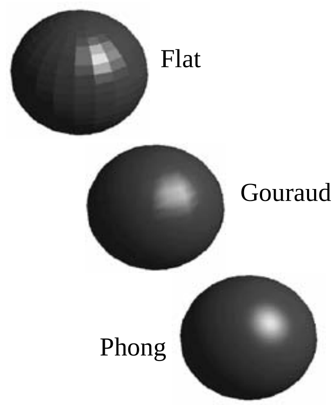
- Ia = (Ys - Y2) / (Y1 - Y2) * I1 + (Y1 - Ys) / (Y1 - Y2) * I2
  Ib = (Ys - Y3) / (Y1 - Y3) * I1 + (Y1 - Ys) / (Y1 - Y3) * I3
  Ip = (Xb - Xp) / (Xb - Xa) * Ia + (Xp - Xa) / (Xb - Xa) * Ib



Gouraud Shading of a sphere

Phong Shading

A more accurate method for rendering a polygon surface is to interpolate normal vectors, and then apply the illumination model to each surface point.

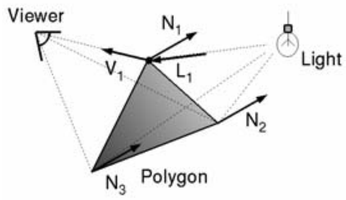


Phong Shading

1. Determine the average unit normal at each polygon vertex.
2. Linearly interpolate the vertex normals over the surface polygon.
3. Apply the illumination model along each scan line to calculate pixel intensities for each surface point.

Phong Shading

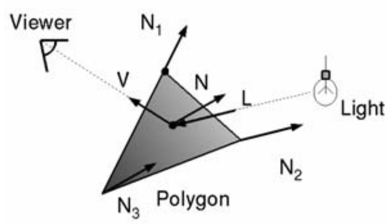
- What if polygonal mesh is too coarse to capture illumination effects in polygon interiors?



$$I = I_E + K_A I_{AL} + \sum_i (K_D (N \bullet L_i) I_i + K_S (V \bullet R_i)^n I_i)$$

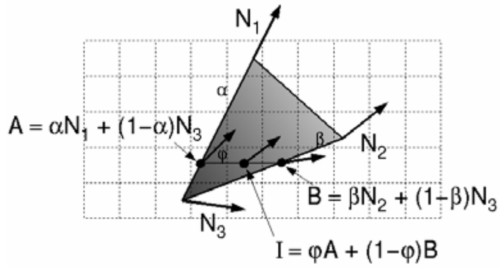
Phong Shading

One lighting calculation per pixel;
Approximate surface normals for points inside polygons by bilinear interpolation of normals from vertices

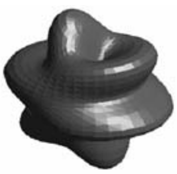


Phong Shading

- Bilinearly interpolate surface normals at vertices down and across scan lines

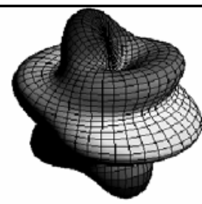


Flat Shading

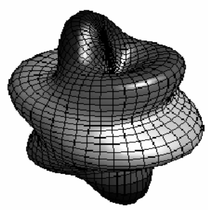


Gouraud Shading

Phong Shading



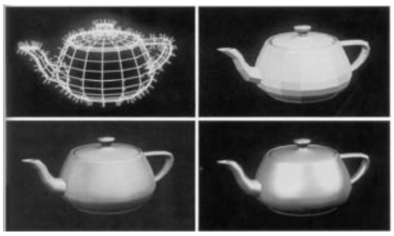
Diffuse surface



With additional specular component

Polygon Shading Algorithms

Wireframe Flat



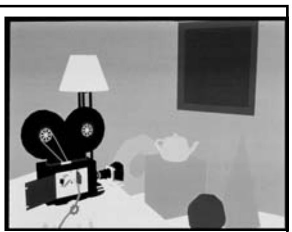
Gouraud Phong

Watt Plate 7

Shading Issues

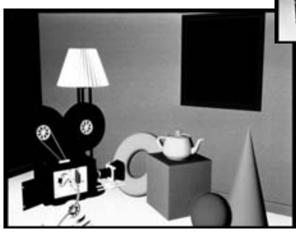
- Problems with interpolated shading:
  - Polygonal silhouettes
  - Perspective distortion
  - Orientation dependence (due to bilinear interpolation)
  - Problems at T-vertices
  - Problems computing shared vertex normals

One shade or color for the entire object, e.g., there really is no shading being done



A Pixar Shutterbug example image with faceted shading.

A Pixar Shutterbug example image with faceted shading.



A Pixar Shutterbug example image with Gouraud shading and no specular highlights.

A Pixar Shutterbug example image with Gouraud shading and no specular highlights.

A Pixar Shutterbug example image with Gouraud shading and specular highlights.

Summary

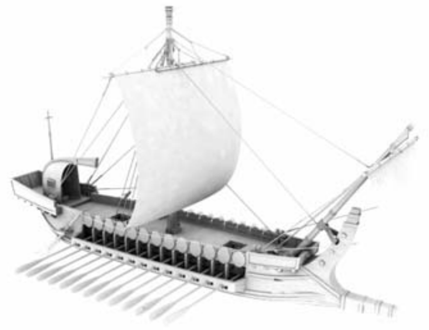
- 2D polygon scan conversion with a sweep-line algorithm
  - Flat
  - Gouraud
  - Phong

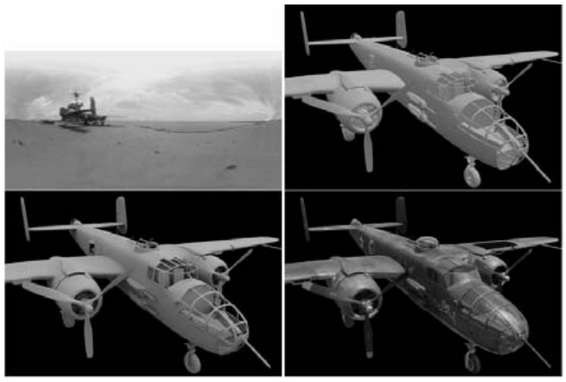
Less expensive

More accurate

Ambient Occlusion

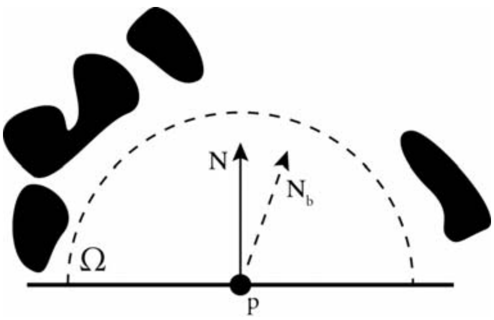
- Full GI still too expensive for full feature film.
- Ambient Occlusion is used in most modern films to simulate indirect lighting in an overcast day.
- Usually, rendered separately and ‘baked’ as texture or 3D data that modifies values of direct lighting.

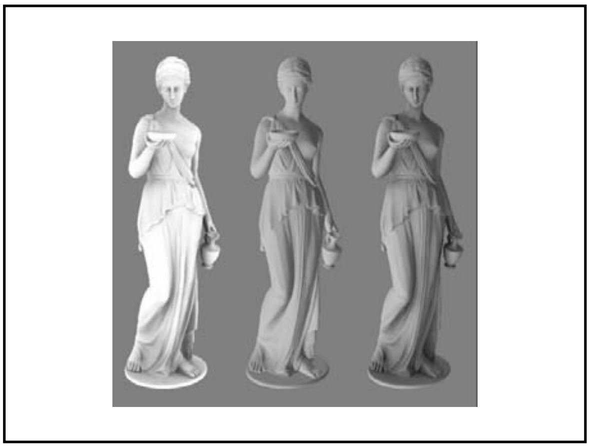


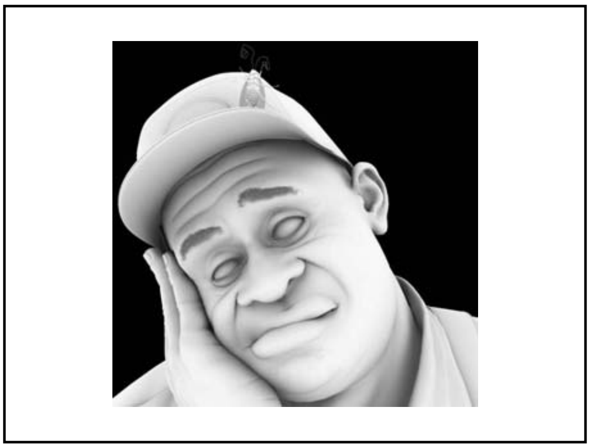


AO - advantages

- Much cheaper than GI.
- Usually does not depend on lighting, looks ok with most light settings.
- Can be computed once for each scene and reused for every frame.





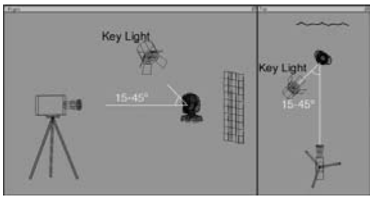



Three Point Lighting

- Basic and commonly used lighting technique
- Key light
- Fill light
- Back light

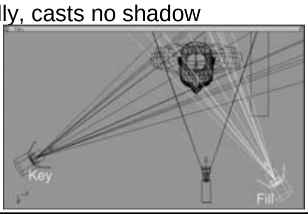
Key light

- Creates the subject's main illumination, and defines the most visible lighting and shadows.
- Simulates main source of illumination



Fill light

- Softens and extends the illumination, simulates secondary light sources
- At most, half as bright as your key light,
- usually, casts no shadow



Back light

- creates a "defining edge" to help visually separate the subject from the background