# CSE-303: COMPUTER GRAPHICS

**PROFESSOR DR. SANJIT KUMAR SAHA**

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAHANGIRNAGAR UNIVERSITY, SAVAR, DHAKA

# COLOR AND SHADING MODELS

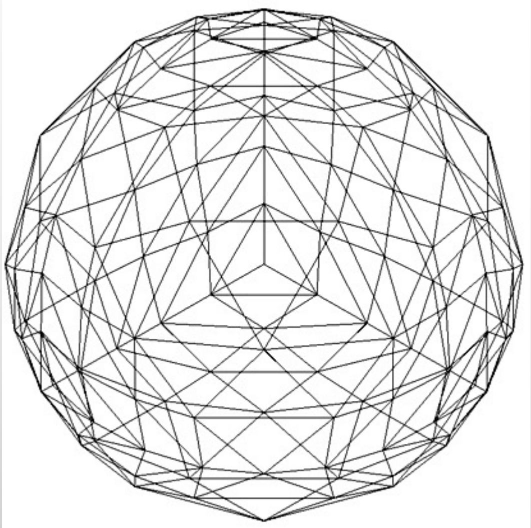
# VIEWING MODELS

**Need for different models**

- **Light models:** to take care of light, its intensity and effect on different textures.
- **Shading models:** to take care of type of surface – its shape, texture etc. And rendering the intensity at every point of the surface.
- **Colour models:** various colour schemes present to represent the calculated intensities on your computer.

# SHADING MODELS

- **Shading models:** we have an illumination model for given point on a surface and every point on the surface is ***unique.***
- Should we calculate intensity of light at all points of the object?
- If our surface is defined as mesh of polygons, which point should be used?



# SHADING MODELS

- Keep in mind:
  - It’s a fairly expensive calculation

  Several possible answers, each with different implications for the visual quality of the result

- Three common techniques are
  - Constant shading (no shading)
  - Faceted shading
  - Gouraud shading
  - Phong shading
- Shading models dictate how often the color computation is performed.

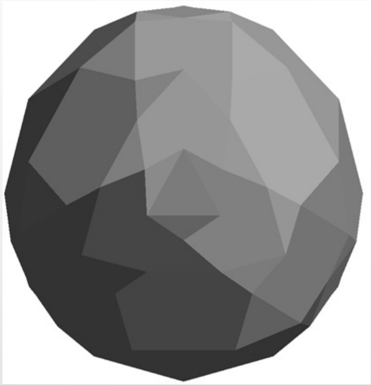
# SHADING METHODS
# POLYGON RENDERING METHODS

## CONSTANT SHADING

*No Shading*

## FACETED SHADING

*Is used for objects defined as polyhedral.*

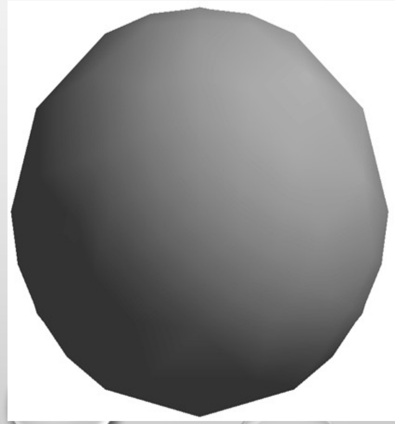


## INTERPOLATION SHADING

*Used for polyhedron-mesh approximation of curved surfaces*

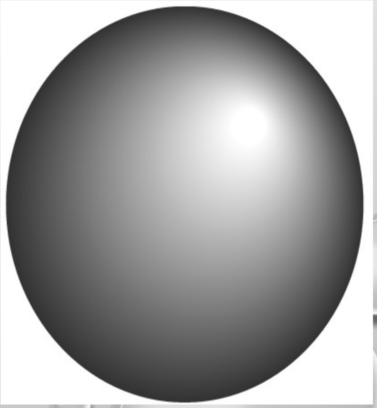
### GOURAUD SHADING

*Interpolate intensity*



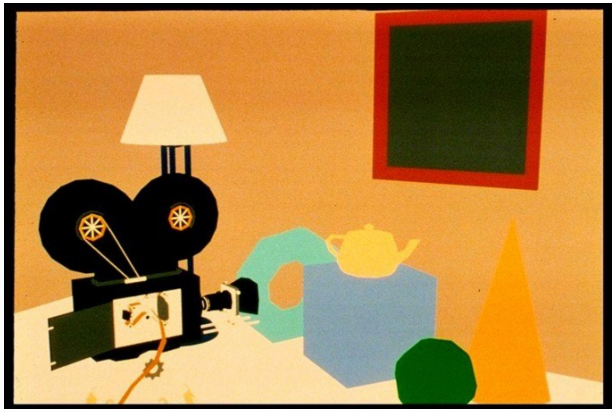
### PHONG SHADING

*Interpolate surface normal*

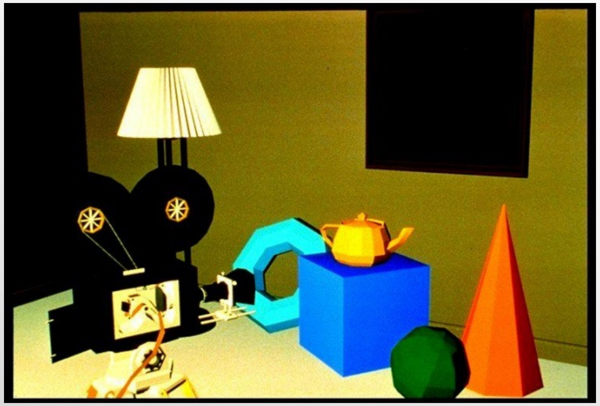


# Effects of different shading techniques

CONSTANT SHADING

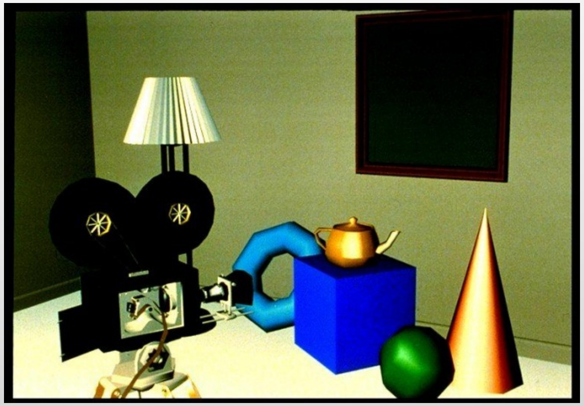


FACETED SHADING

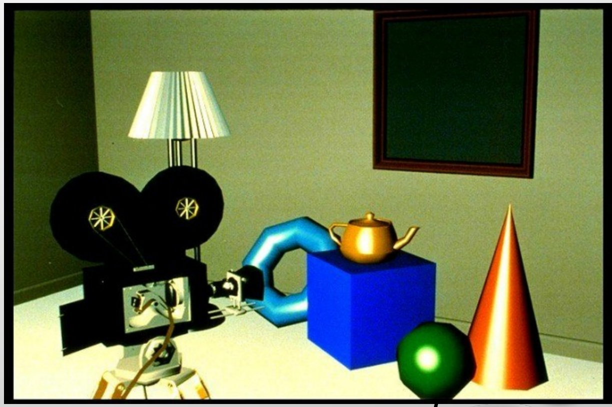


INTERPOLATION SHADING

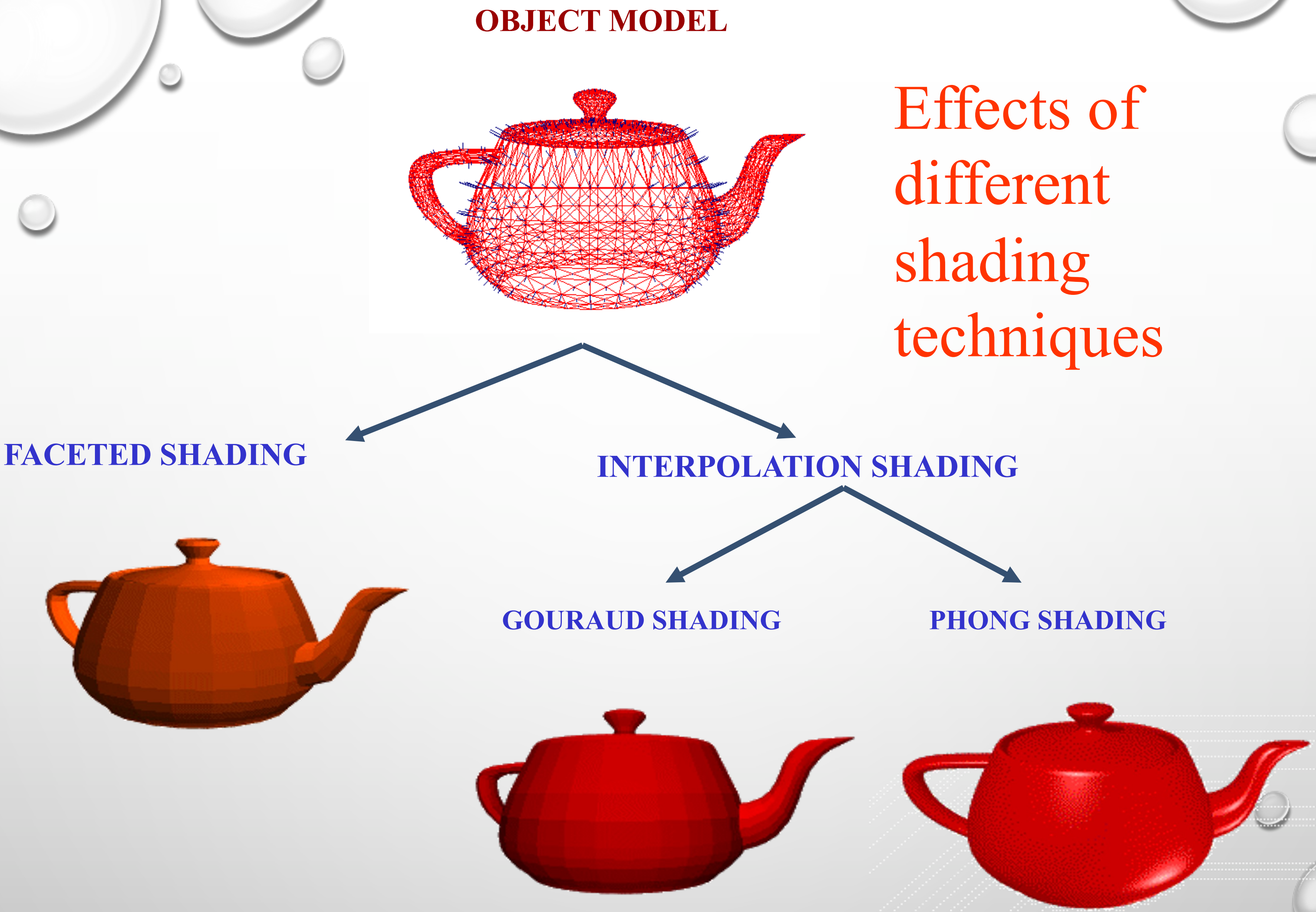
GOURAUD SHADING



PHONG SHADING



# Effects of different shading techniques



OBJECT MODEL

FACETED SHADING

INTERPOLATION SHADING

GOURAUD SHADING

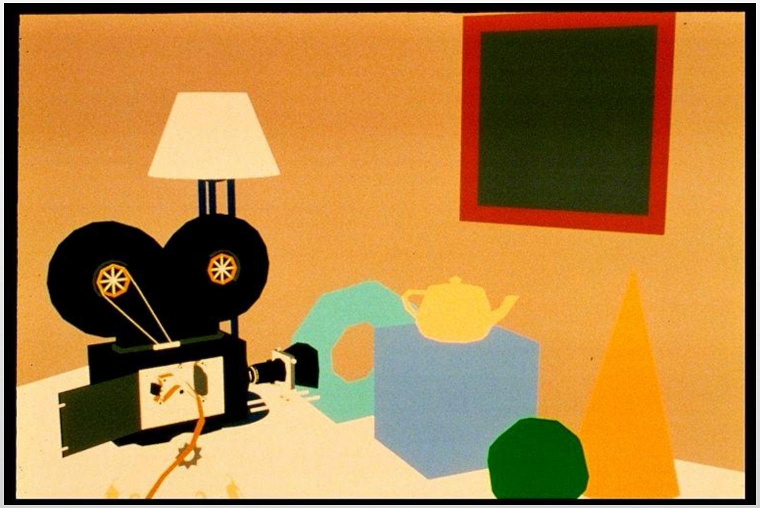
PHONG SHADING

# SHADING MODELS

- We will start to look at shading or rendering methods commonly used in computer graphics
    - Constant surface shading/rendering
    - Faceted surface shading/rendering
    - Gouraud surface shading/rendering
    - Phong surface shading/rendering

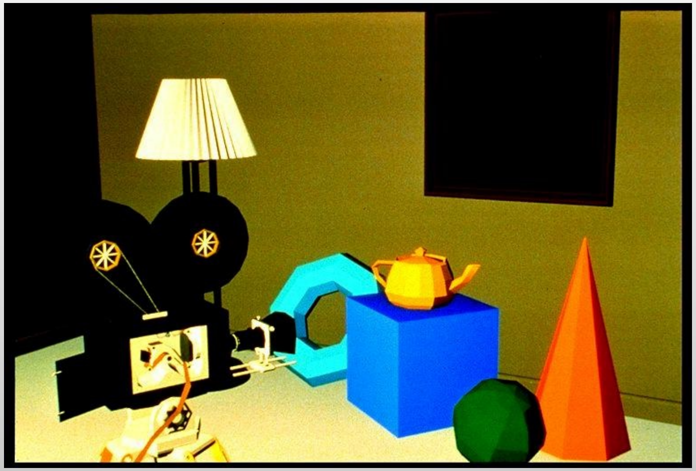
# CONSTANT INTENSITY SHADING

- Constant intensity shading also refer to as no shading.
- It is done by giving same color to every polygon in the object.
- Thus entire object is painted with the same color.
- It is the simplest, but not realistic.

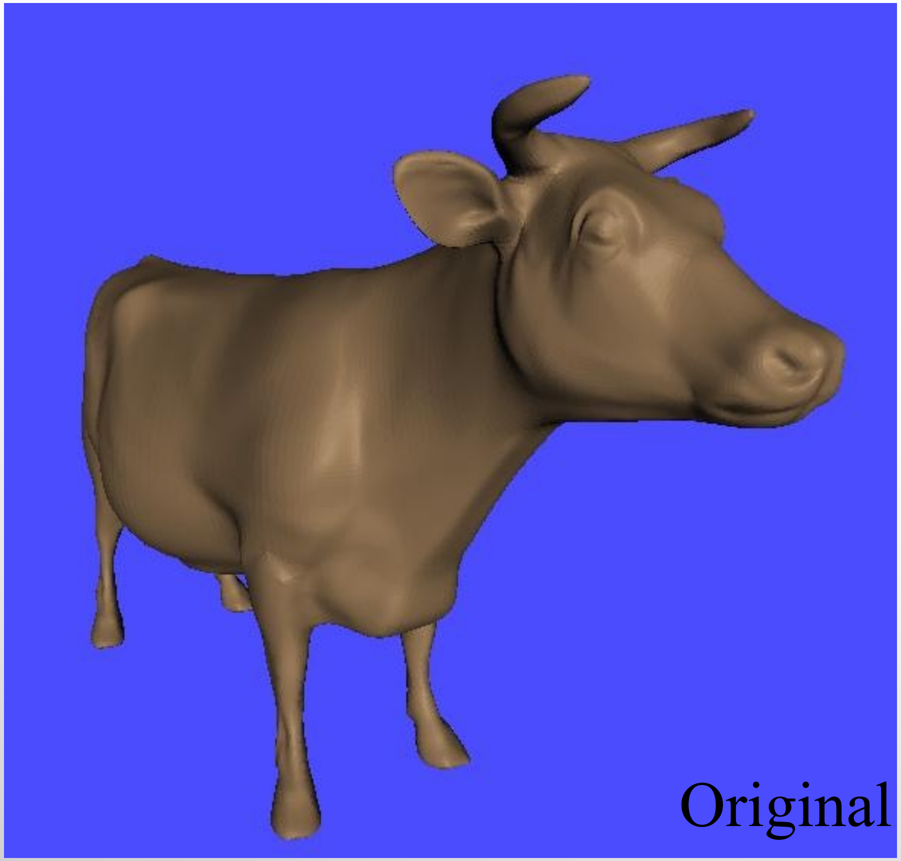


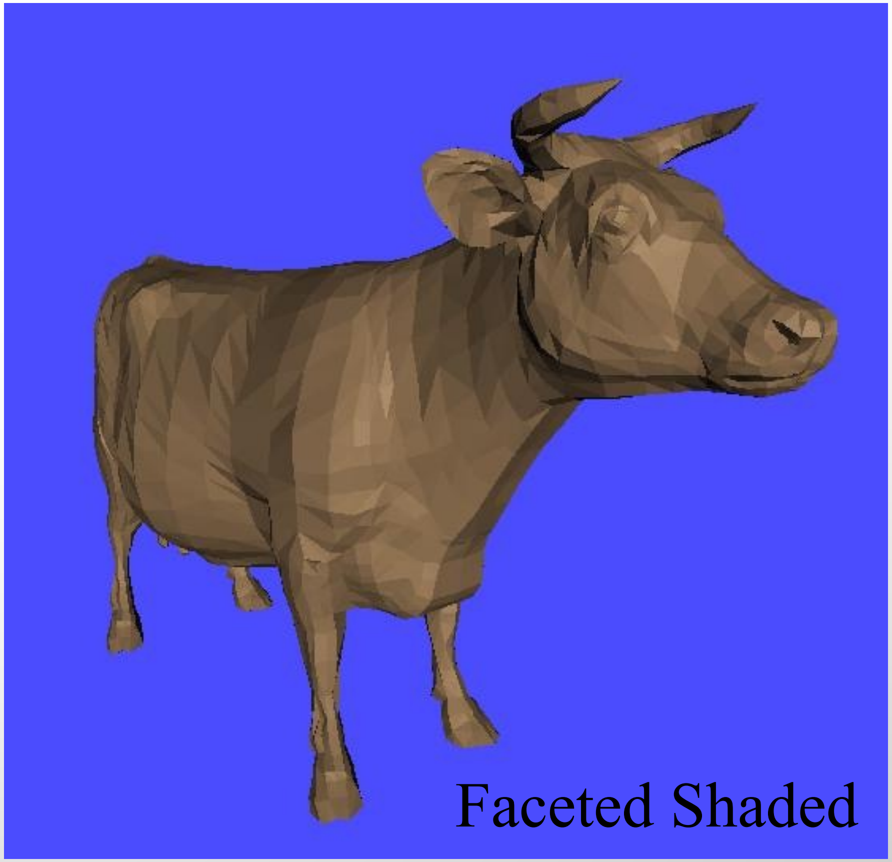
# FACETED SHADING

- Also known as ***flat shading or fast shading*** it is an extension to constant intensity shading.
- The entire object is divided into number of polygons surfaces called ***facet***s.
- Determine the orientation of each facet and does one color computation for each polygon or facet.
- It is simple, fast (?) and produce better results than no shading.



# FACETED SHADING





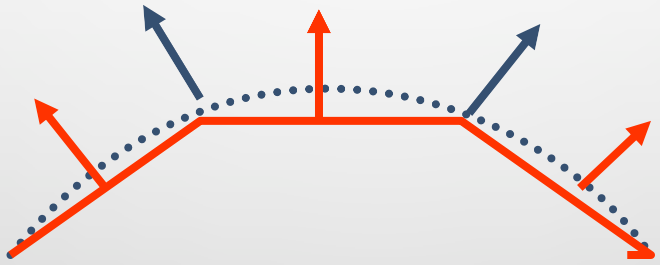
- Just add lots and lots of polygons – however, this is SLOW!

# FACETED SHADING

- Still it is not realistic as
  - For point sources, the direction to light varies across the facet
  - For specular reflectance, direction to eye varies across the facet
- We have to determine the orientation of each facet we have to find the  normal for each facet. This approximation effect the result.

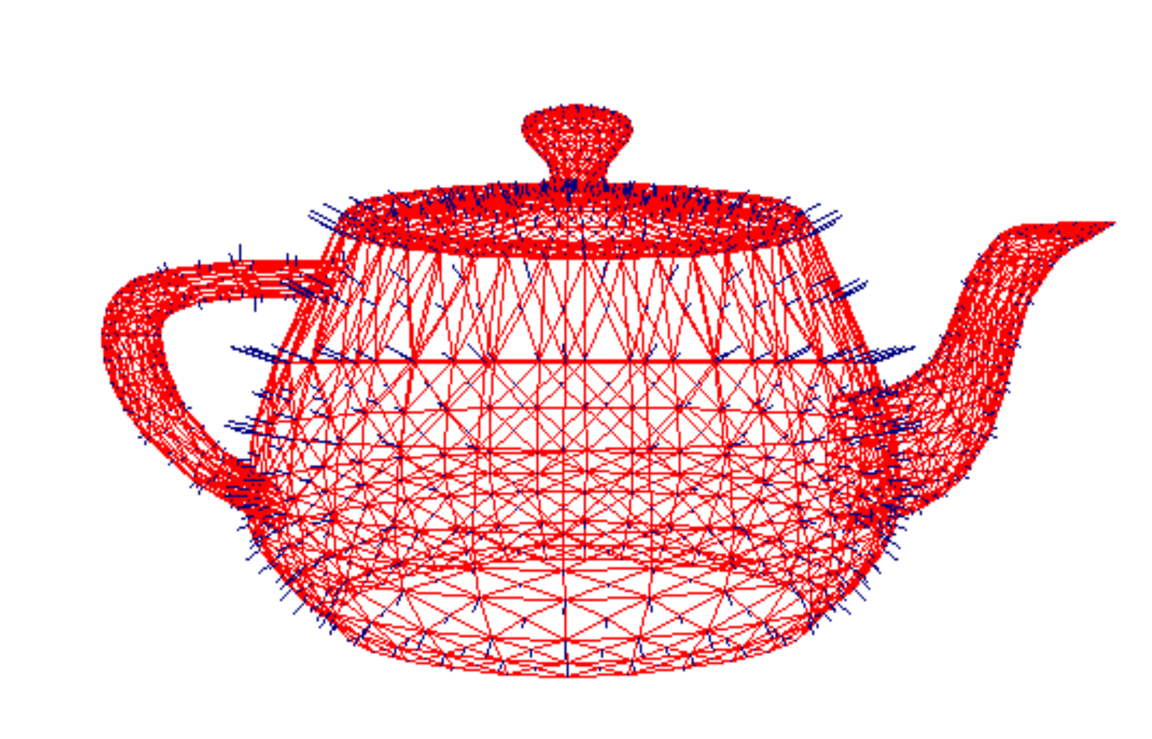
# FACETED SHADING

- To determine the orientation of each facet we introduce *vertex normal* at each vertex
  - Usually different from facet normal
  - Used *only* for shading
  - Think of as a better approximation of the *real* surface that the polygons approximate



# FACETED SHADING

- Vertex normals may be
  - Approximated by averaging the normals of the facets that share the vertex
  - Provided with the model

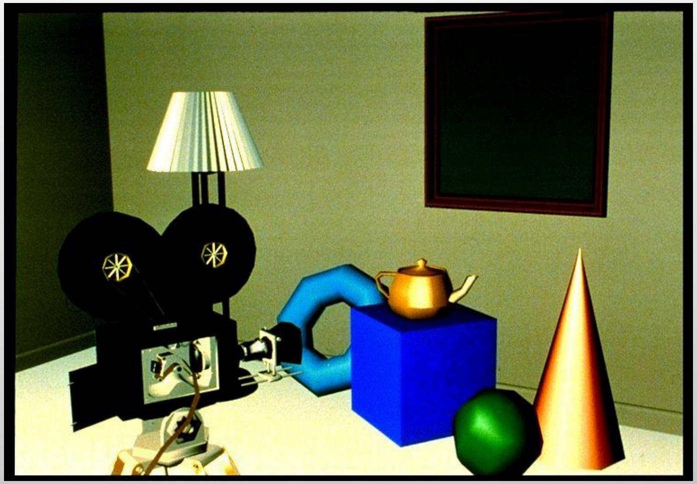


# FACETED SHADING

- It is very useful for quickly displaying the general appearance of surfaces
- It is more applicable when
  - The object must be polyhedron and not an approximation of an object with a curved surface.
  - All light sources illuminating the object are sufficiently far from the surface so that n.l and the attenuation functions are constant over the surface.
  - The viewing position is sufficiently far from the surface so that v.r is constant over the surface.

# GOURAUD SHADING

- Gouraud surface shading was developed in the 1970s by Henri Gouraud while working at the University of Utah along with Ivan Sutherland and David Evans
- Often also called **intensity-interpolation surface rendering**
- This is the most common approach
  - Intensity levels are calculated at each vertex
  - Linearly interpolate the resulting colors over faces
    - Along edges
    - Along scanlines

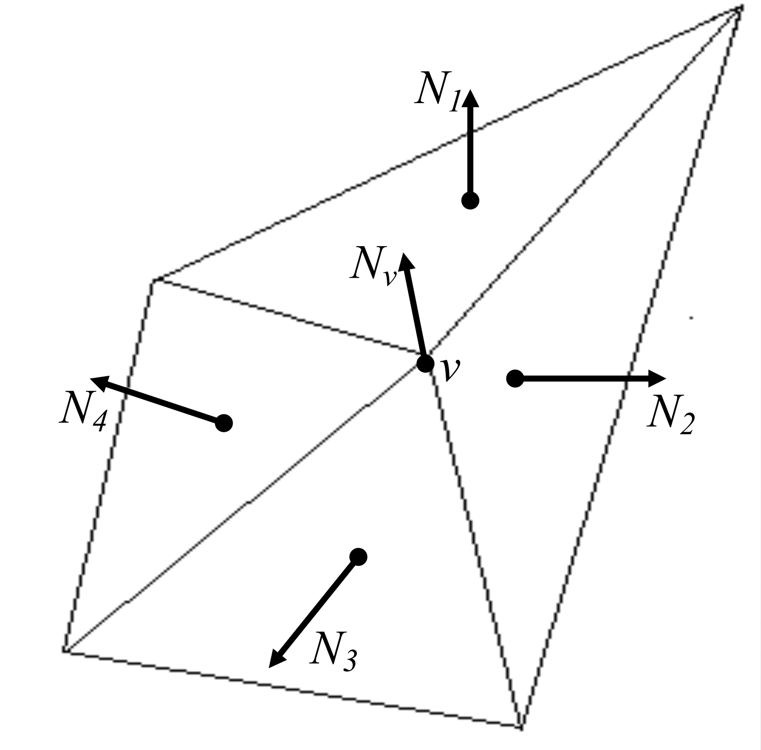


# GOURAUD SHADING

- To render a polygon, gouraud surface rendering proceeds as follows:
  1. Determine the average unit normal vector at each vertex of the polygon
  2. Apply an illumination model at each polygon vertex to obtain the light intensity at that position
  3. Linearly interpolate the vertex intensities over the projected area of the polygon

# GOURAUD SHADING

Step 1: Determine the average unit normal vector at each vertex of the polygon



- The average unit normal vector at *v* is given as:

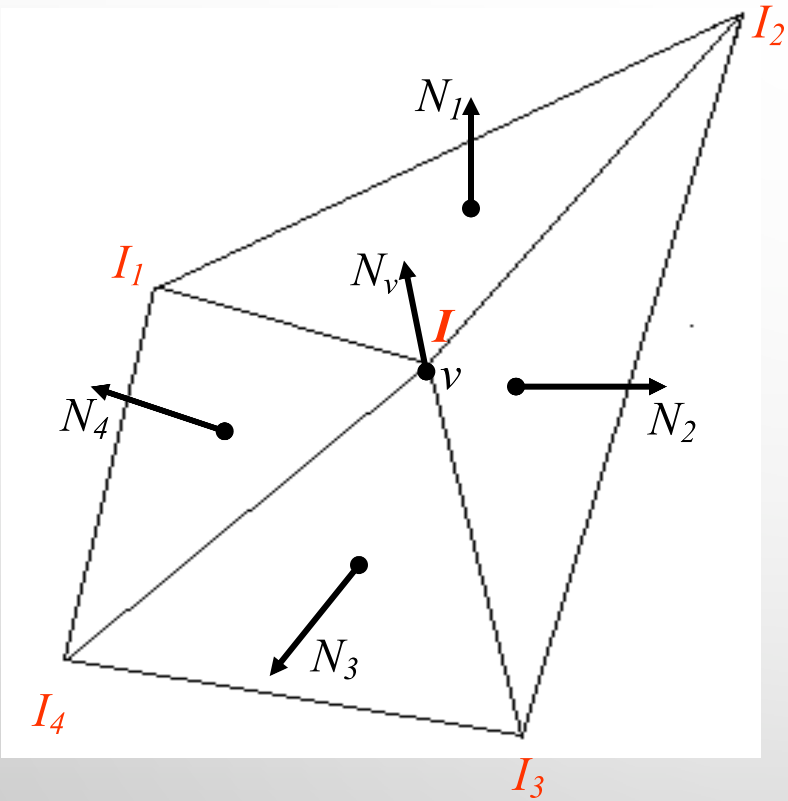
$$N_v = \frac{N_1 + N_2 + N_3 + N_4}{\left|N_1 + N_2 + N_3 + N_4\right|}$$

- Or more generally:

$$N_v = \frac{\sum_{i=1}^{n} N_i}{\left|\sum_{i=1}^{n} N_i\right|}$$

# GOURAUD SHADING

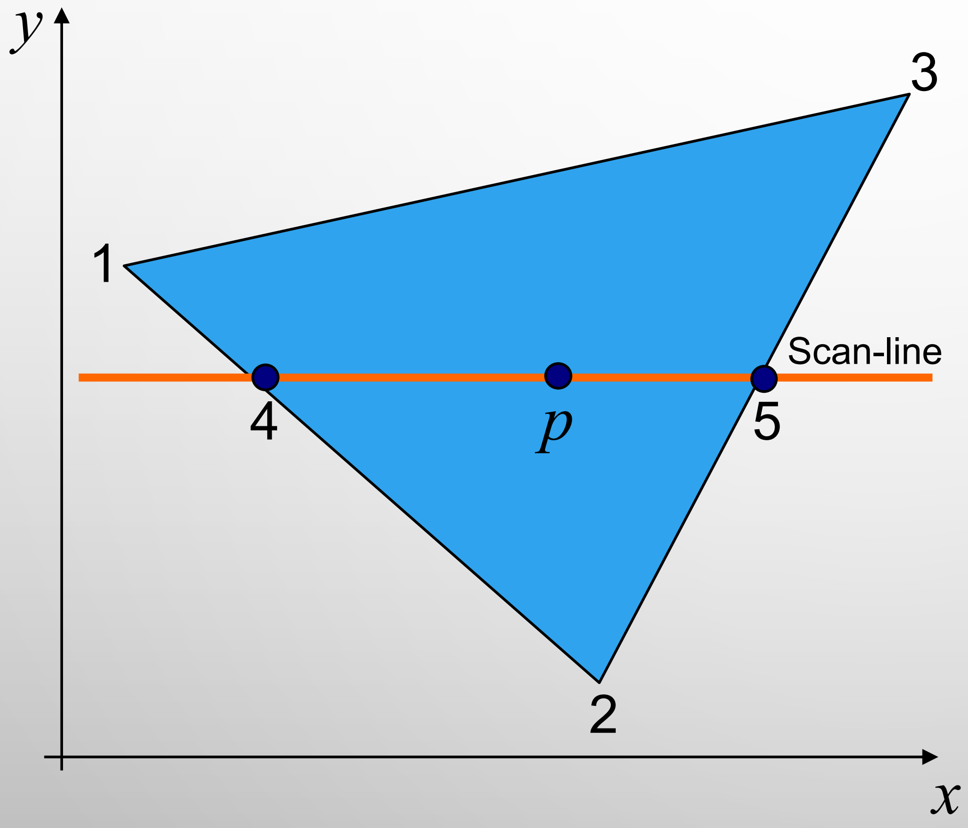
Step 2: Apply an illumination model at each polygon vertex



- You have to apply the illumination model to take of
  - Ambient light
  - Diffuse reflection
  - Specular reflection
  - Transparency (refraction)

# GOURAUD SHADING

Step 3: Linearly interpolate the vertex intensities over the projected area of the polygon along edges and along scan lines

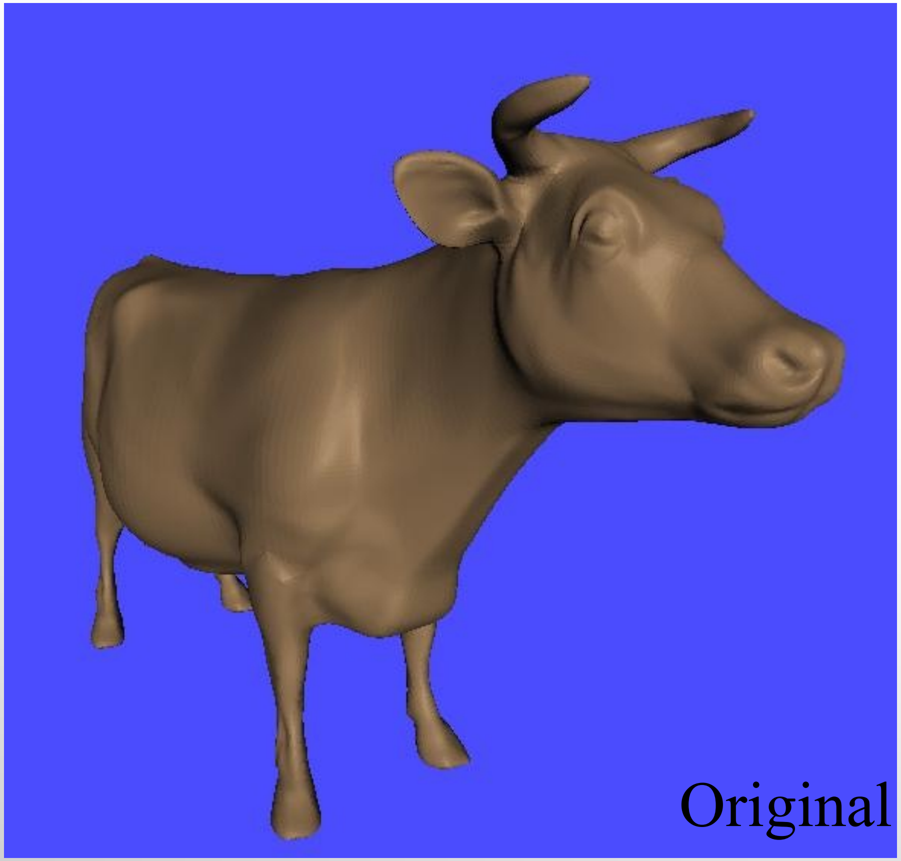


$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

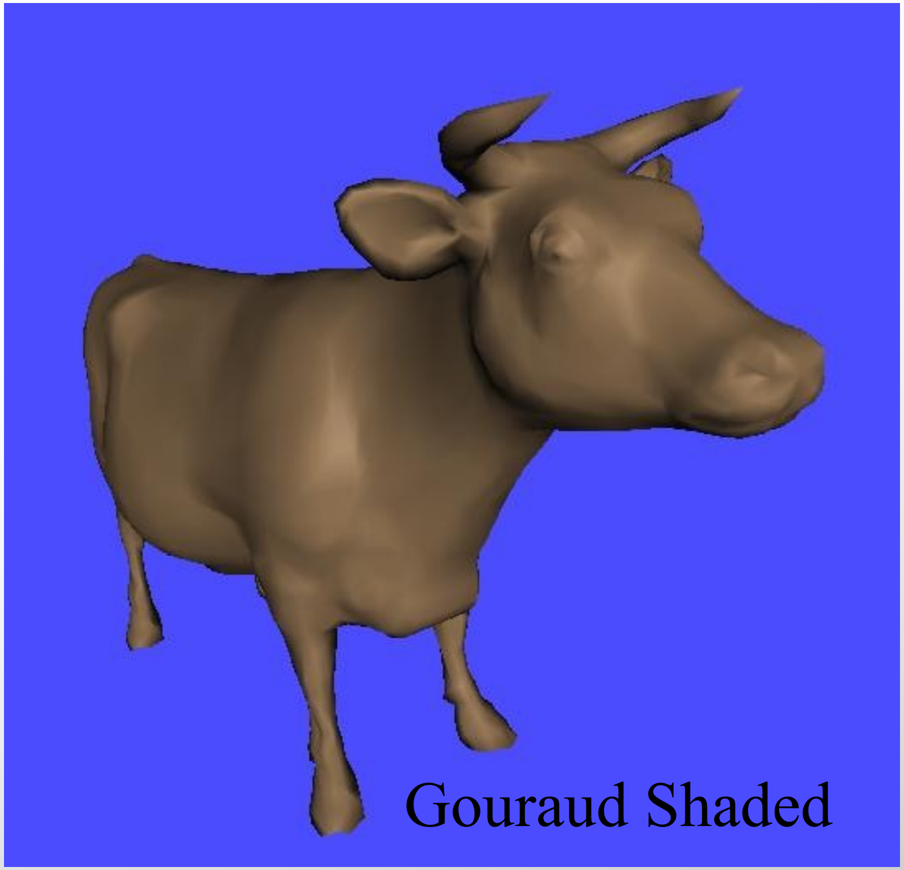
$$I_5 = \frac{y_5 - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_5}{y_3 - y_2} I_2$$

$$I_p = \frac{x_5 - x_p}{x_5 - x_4} I_4 + \frac{x_p - x_4}{x_5 - x_4} I_5$$

# GOURAUD SHADING



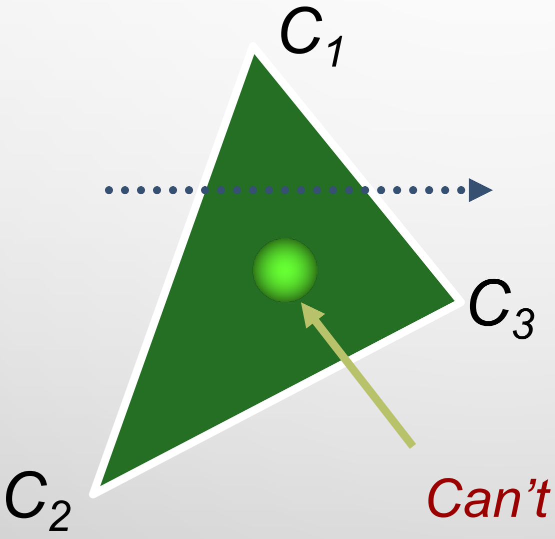
Original

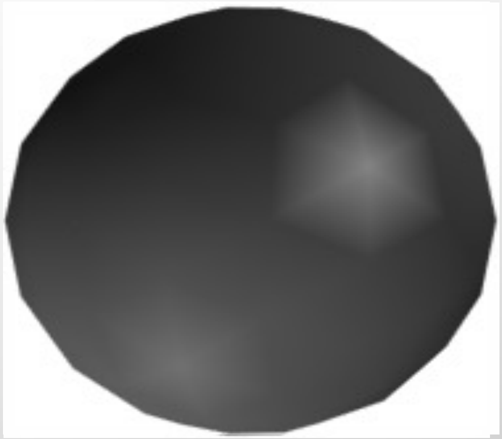


Gouraud Shaded

# GOURAUD SHADING

- Objects shaded with gouraud shading often appears dull, chalky
- Lacks accurate specular component
  - If included, will be averaged over entire polygon

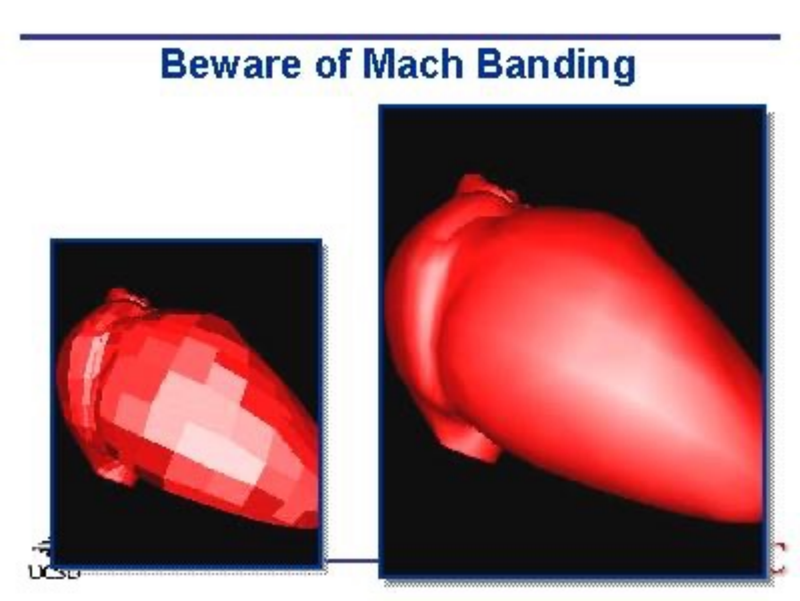




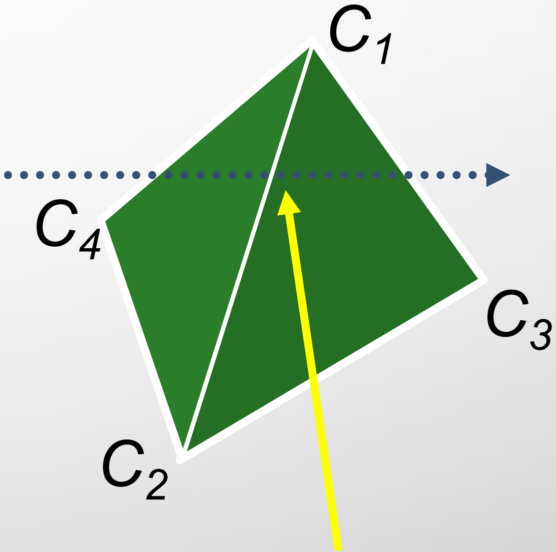
*Can't shade that effect!*

# GOURAUD SHADING

- Gouraud shading can introduce anomalies known as **mach bands**
- **Mach banding**
  - Artifact at discontinuities in intensity or intensity slope



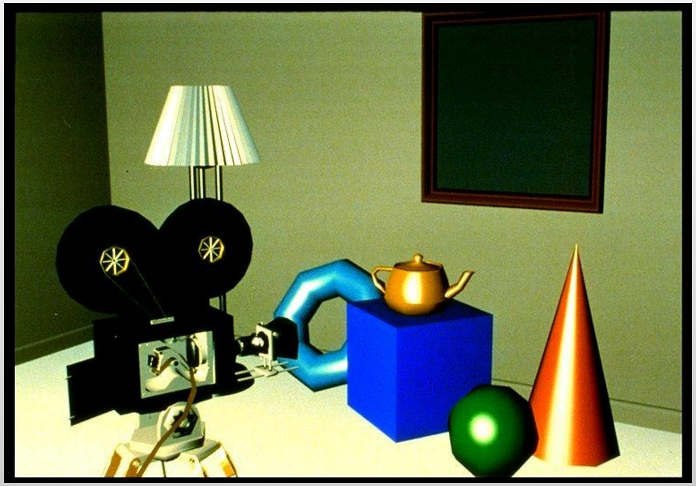
Beware of Mach Banding



*Discontinuity in rate of color change occurs here*

# PHONG SHADING

- A more accurate interpolation based approach for rendering a polygon was developed by Phong Bui Tuong
- Basically the phong surface rendering model interpolates normal vectors instead of intensity values and then compute the actual intensity value using ***phong lighting model***.
- Often also called **normal or vector – interpolation surface rendering**



# PHONG SHADING

- ***Phong shading*** is not the same as phong lighting, though they are sometimes mixed up
  - Phong lighting: the empirical model we've been discussing to calculate illumination at a point on a surface
  - Phong shading: linearly interpolating the surface normal across the facet, applying the phong lighting model at every pixel

$$I_{total} = k_a I_{ambient} + \sum_{i=1}^{\#lights} I_i \left( k_d \left( \hat{N} \cdot \hat{L}_i \right) + k_s \left( \hat{V} \cdot \hat{R}_i \right)^{n_{shiny}} \right)$$

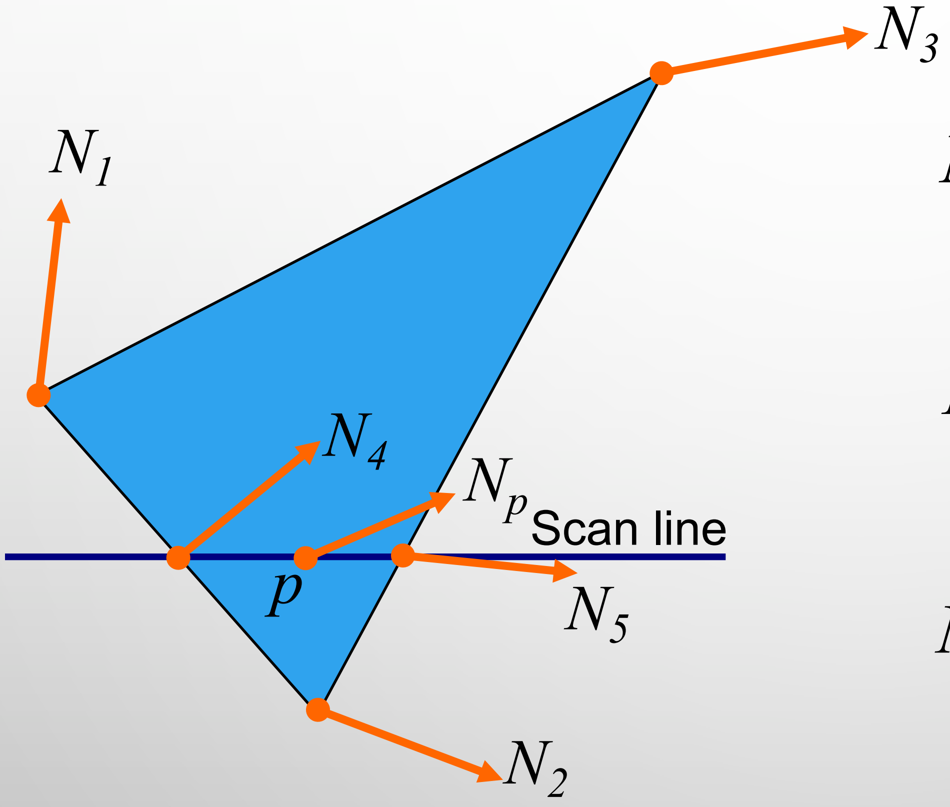
# PHONG SHADING

- To render a polygon, phong surface rendering proceeds as follows:

1. Determine the average unit normal vector at each vertex of the polygon
2. Linearly interpolate the vertex normals over the projected area of the polygon
3. Apply an illumination model at positions along scan lines to calculate pixel intensities using the interpolated normal vectors

# PHONG SHADING

- Interpolating normal



$$N_4 = \frac{y_4 - y_2}{y_1 - y_2} N_1 + \frac{y_1 - y_4}{y_1 - y_2} N_2$$

$$N_5 = \frac{y_5 - y_2}{y_3 - y_2} N_3 + \frac{y_3 - y_5}{y_3 - y_2} N_2$$

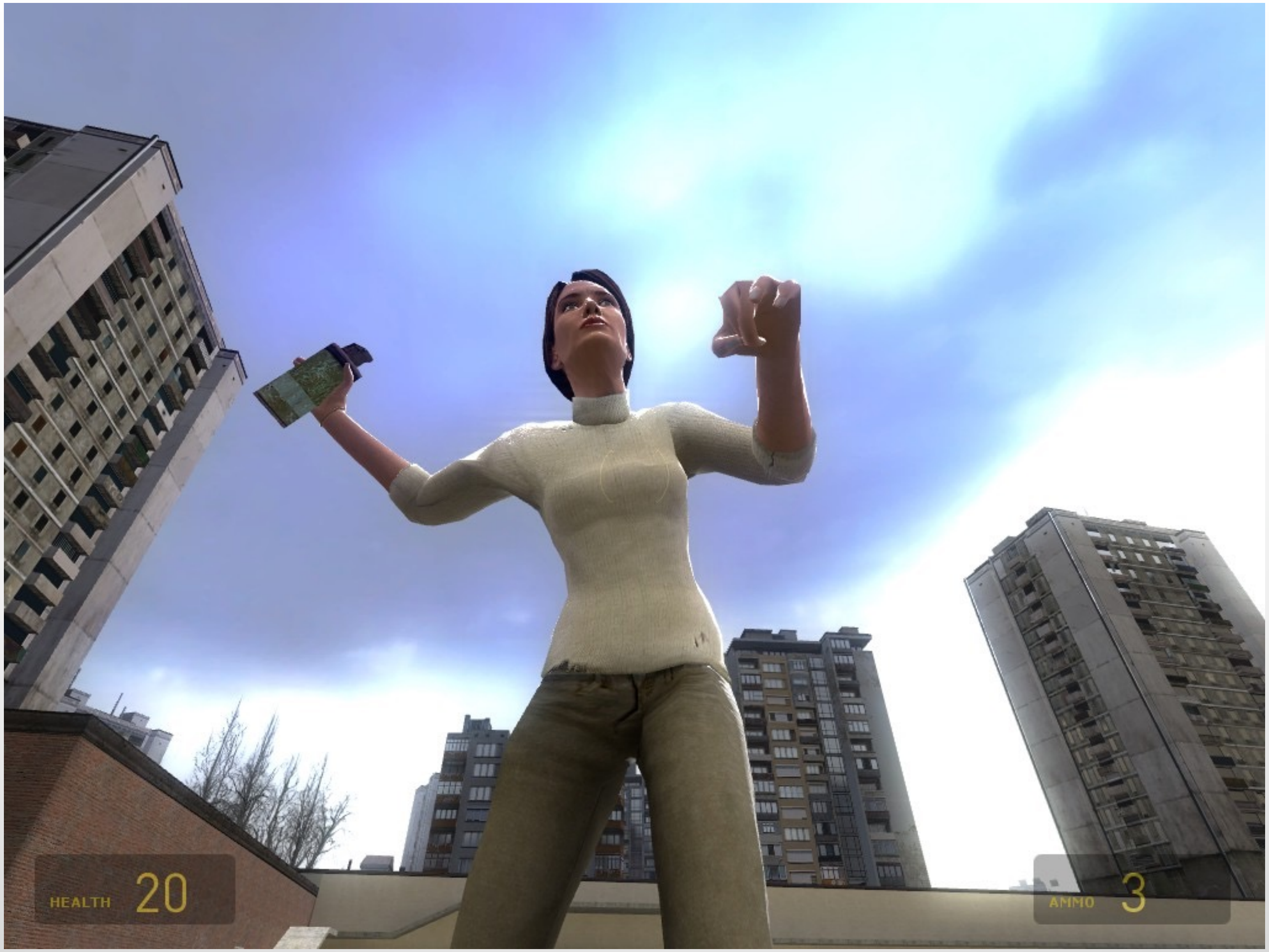
$$N_p = \frac{y_p - y_5}{y_4 - y_5} N_4 + \frac{y_4 - y_p}{y_4 - y_5} N_5$$

# PHONG SHADING

- Phong shading accept the same input as gouraud shading and produce very smooth-looking results.
- Phong shading is considerably more expensive.
- Phong shading is much slower than gouraud shading as the lighting model is revaluated so many times
- However, there are fast phong surface rendering approaches that can be implemented iteratively.

# PHONG SHADING EXAMPLES

# PHONG SHADING EXAMPLES



# SUMMARY

- For realistic rendering of polygons we need interpolation methods to determine lighting positions
- Flat shading is fast, but unrealistic
- Gouraud shading is better, but does not handle specular reflections very well
- Phong shading is better still, but can be slow