Dr. George Karraz, Ph. D.

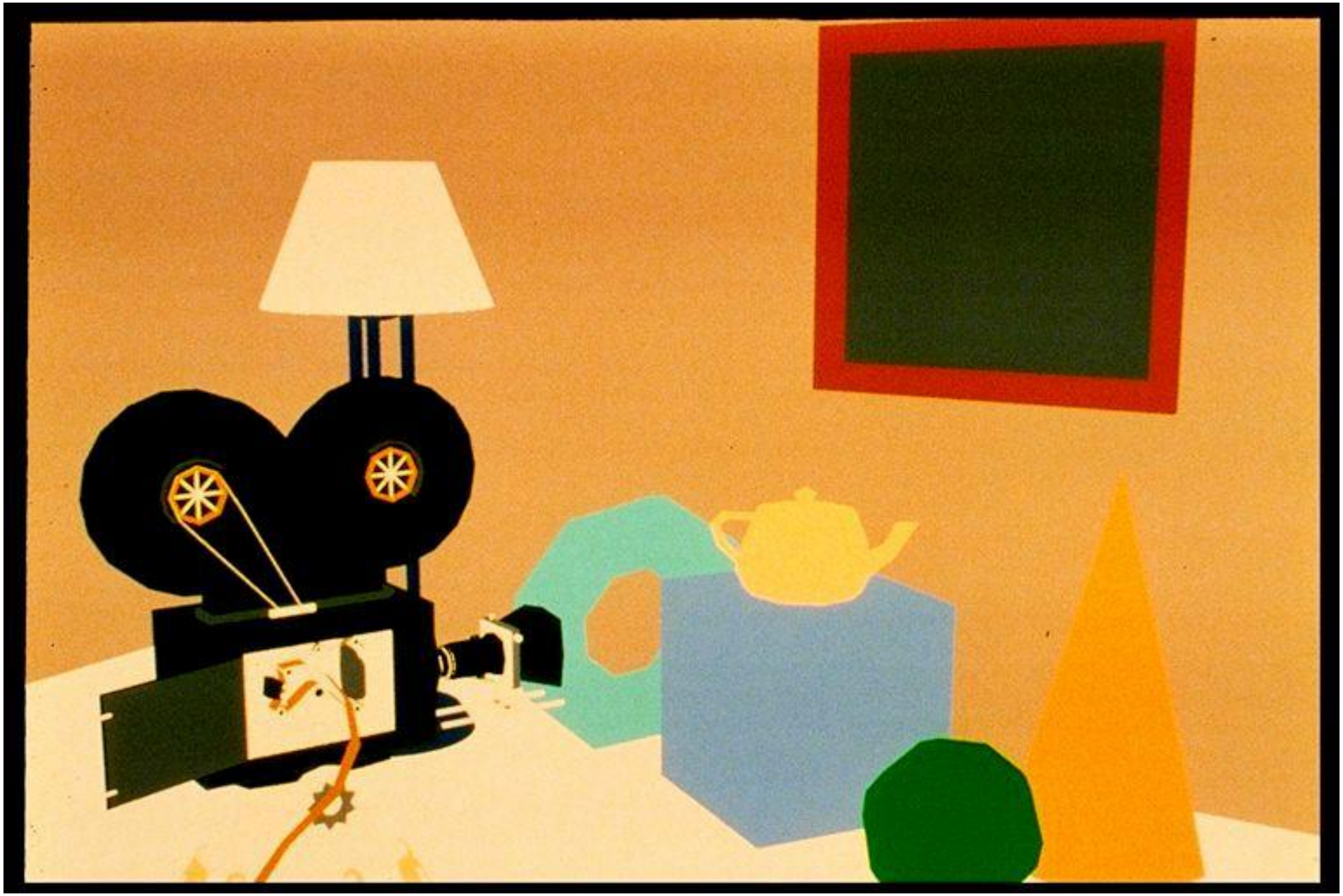
Polygon Rendering Methods

Contents

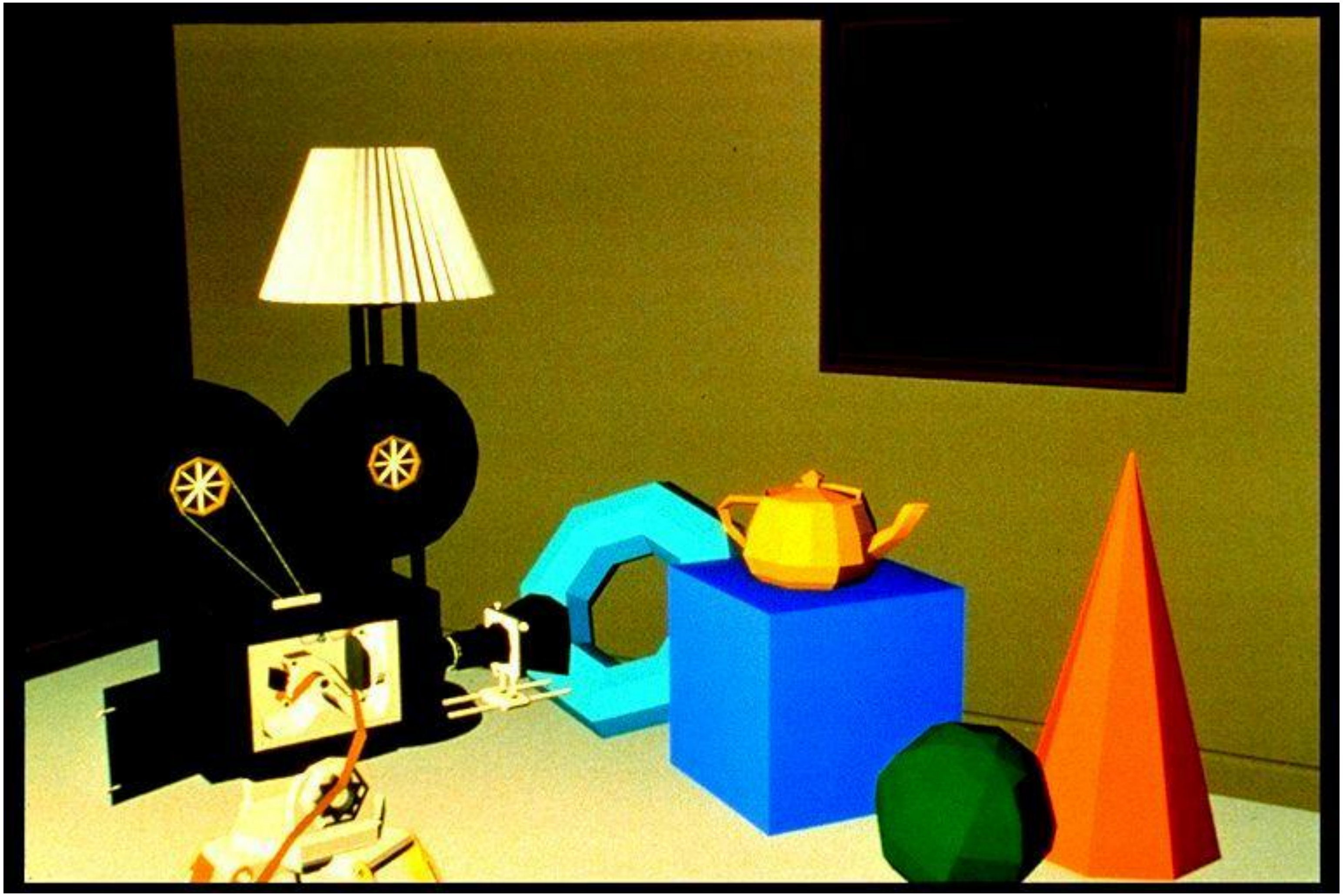
Today we will start to look at rendering methods used in computer graphics

- Flat surface rendering
- Gouraud surface rendering
- Phong surface rendering

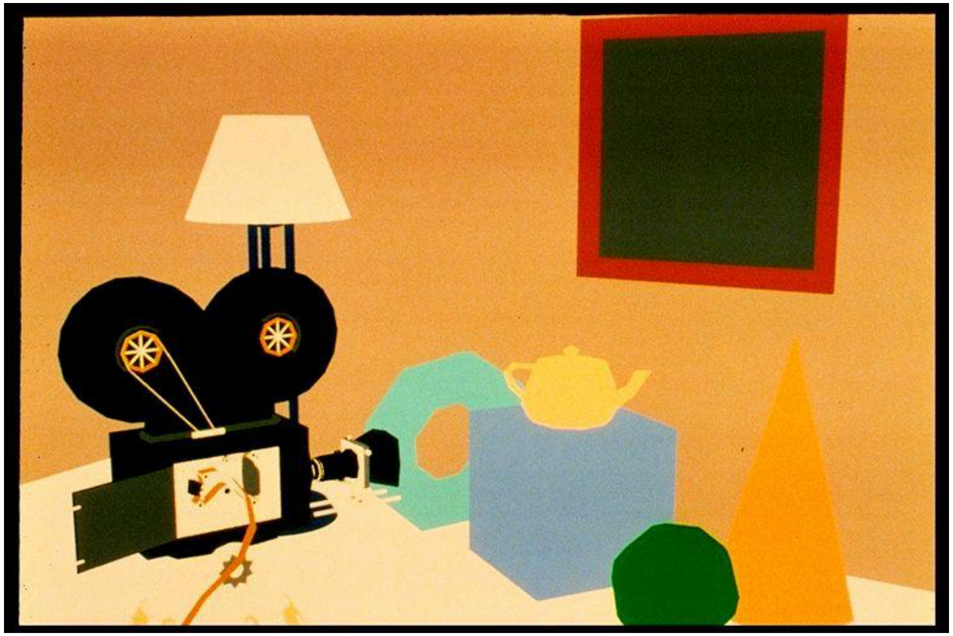
No Surface Rendering



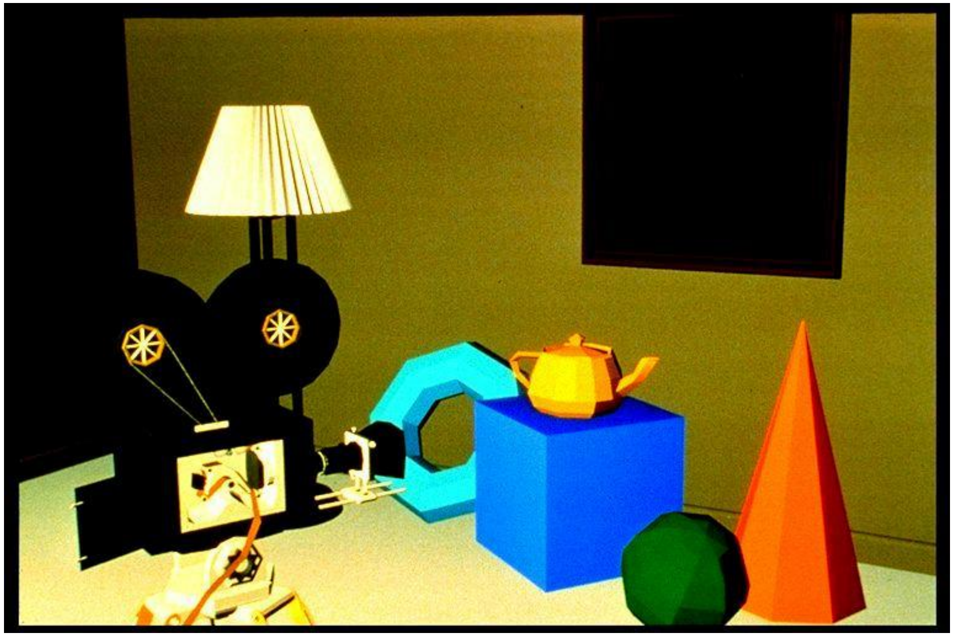
Flat Surface Rendering



No Surface Rendering Vs Flat Surface Rendering

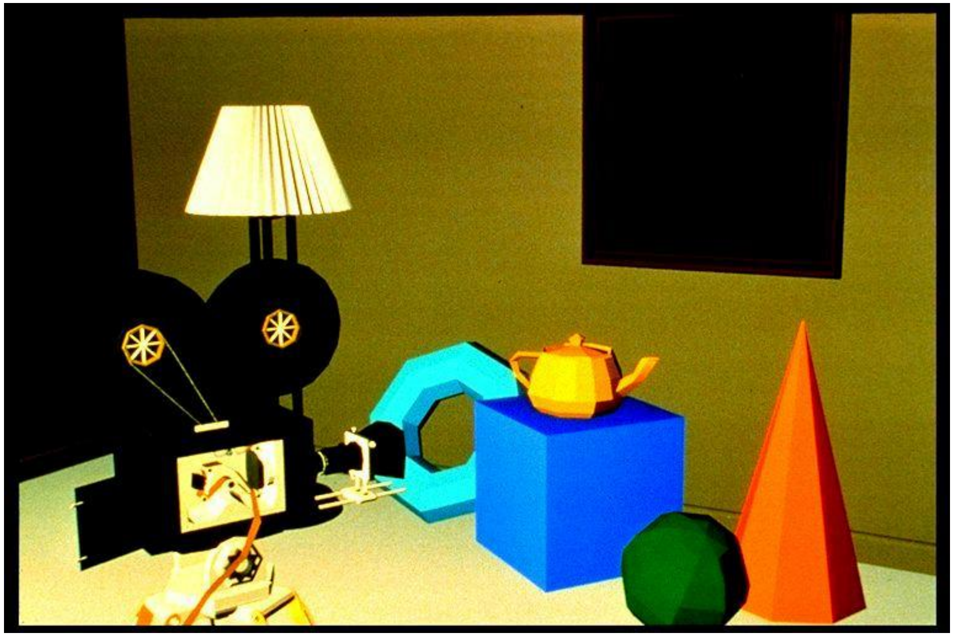


No Surface Rendering

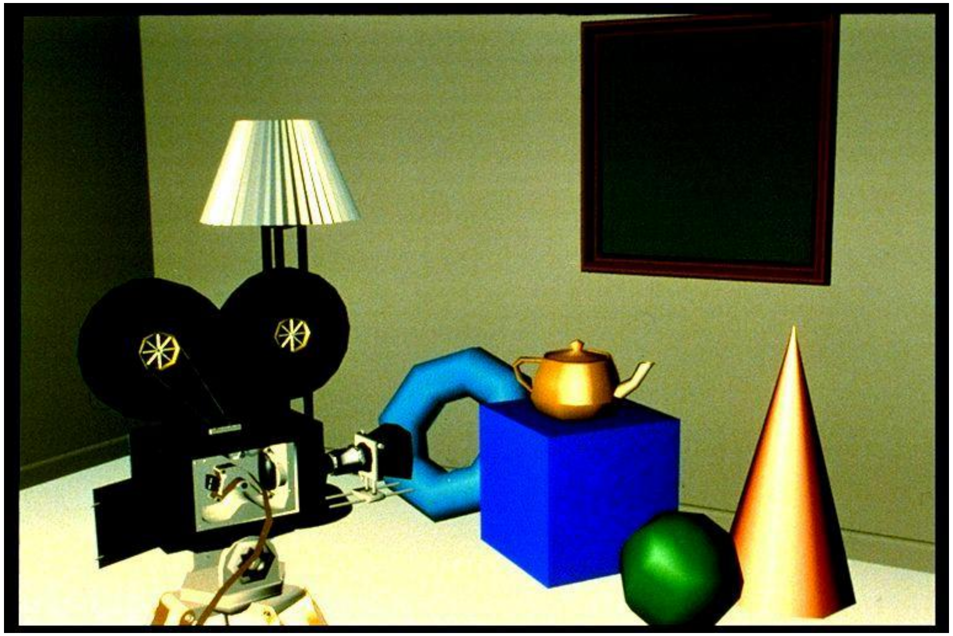


Flat Surface Rendering

Gouraud Surface Rendering Vs. Flat Surface Rendering

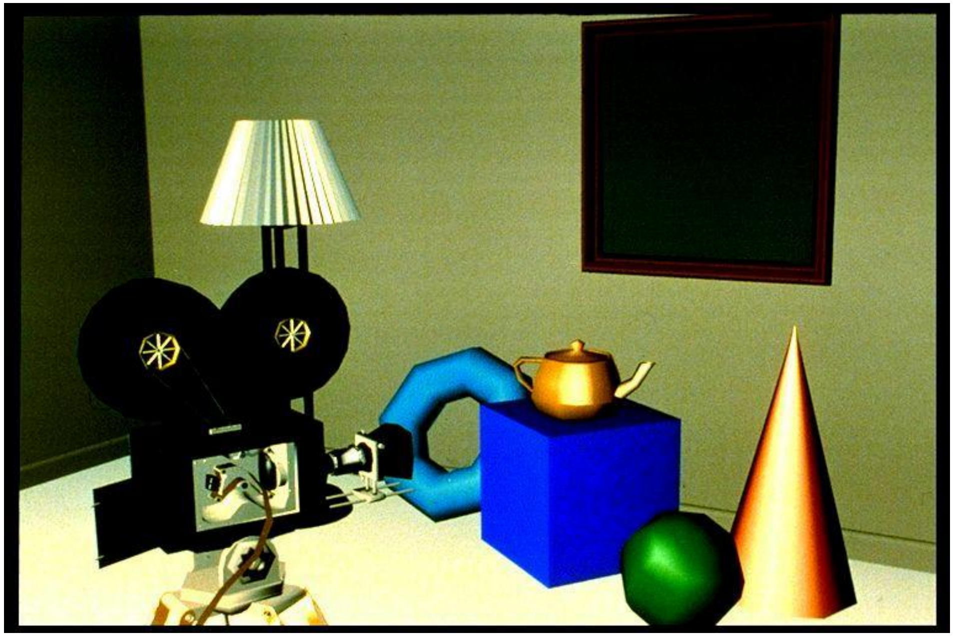


Flat Surface Rendering

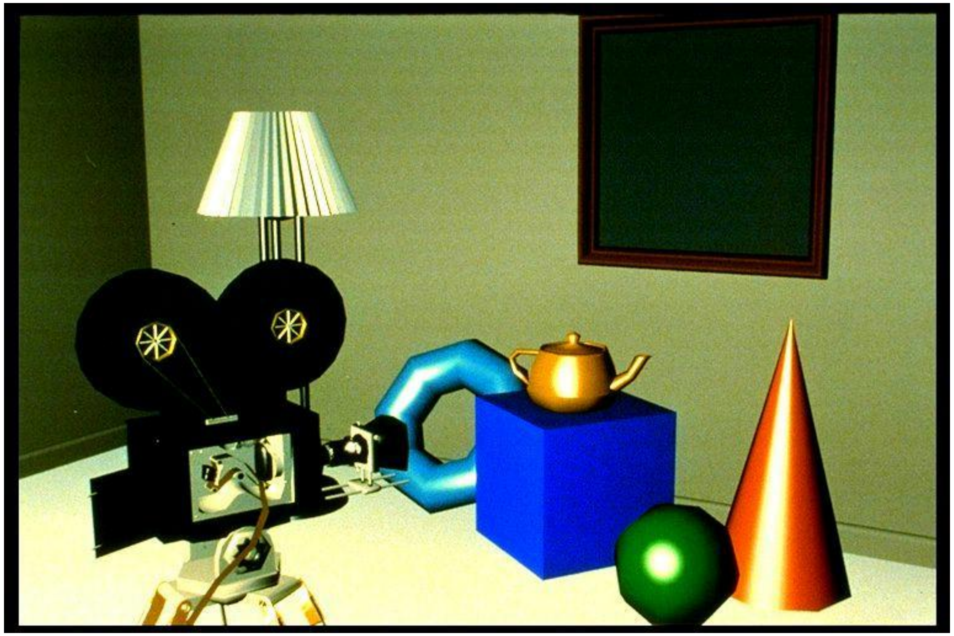


Gouraud Surface Rendering

Gouraud Surface Rendering Vs Phong Surface Rendering



Gouraud Surface Rendering



Phong Surface Rendering

Flat Surface Rendering

The simplest method for rendering a polygon surface

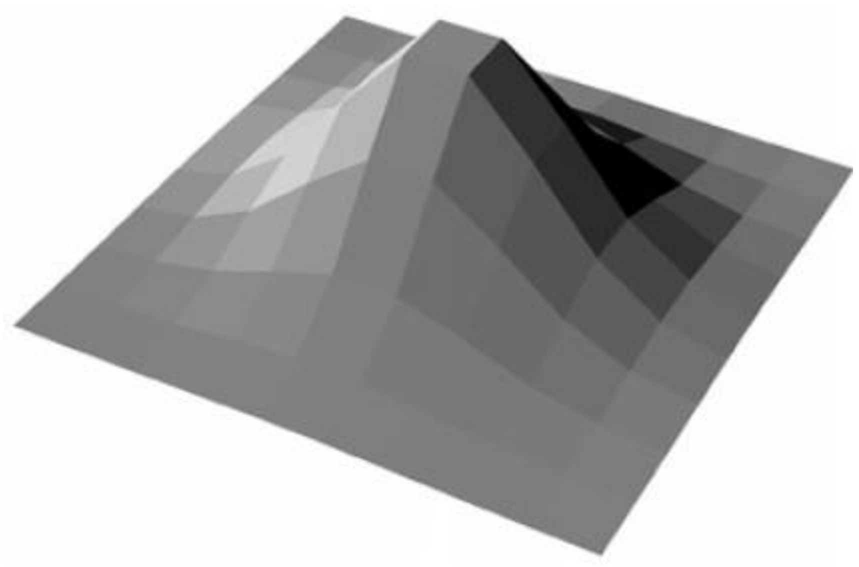
The same colour is assigned to all surface positions

The illumination at a single point on the surface is calculated and used for the entire surface

Flat surface rendering is extremely fast, but can be unrealistic

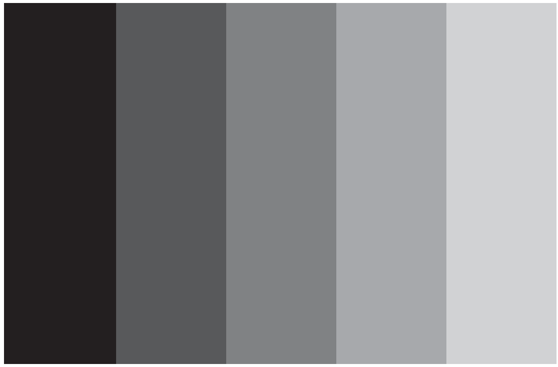
Flat Shading

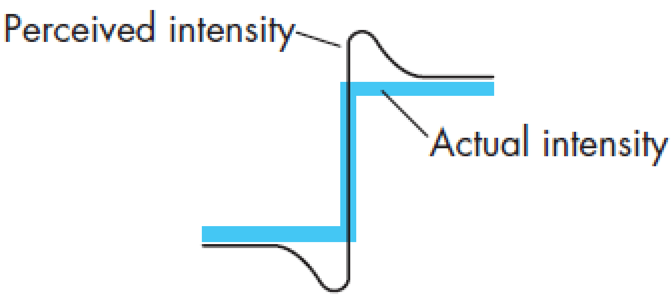
If our polygonal mesh has been designed to model a smooth surface, flat shading will almost always be disappointing because we can see even small differences in shading between adjacent polygons



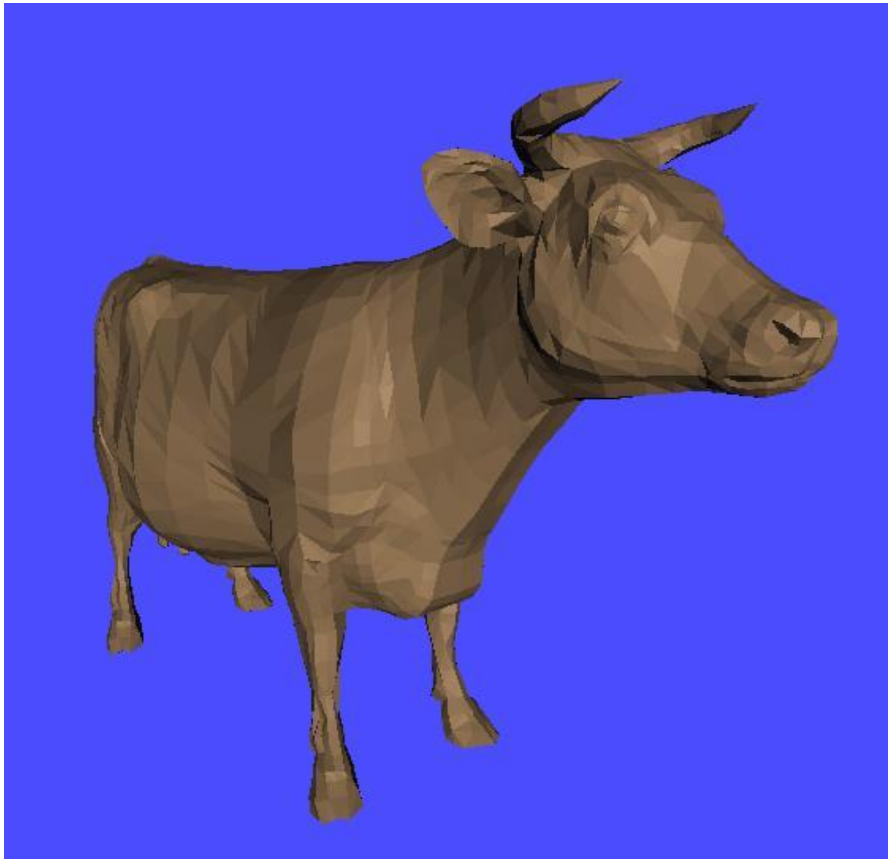
Flat Shading Problems: Mach bands

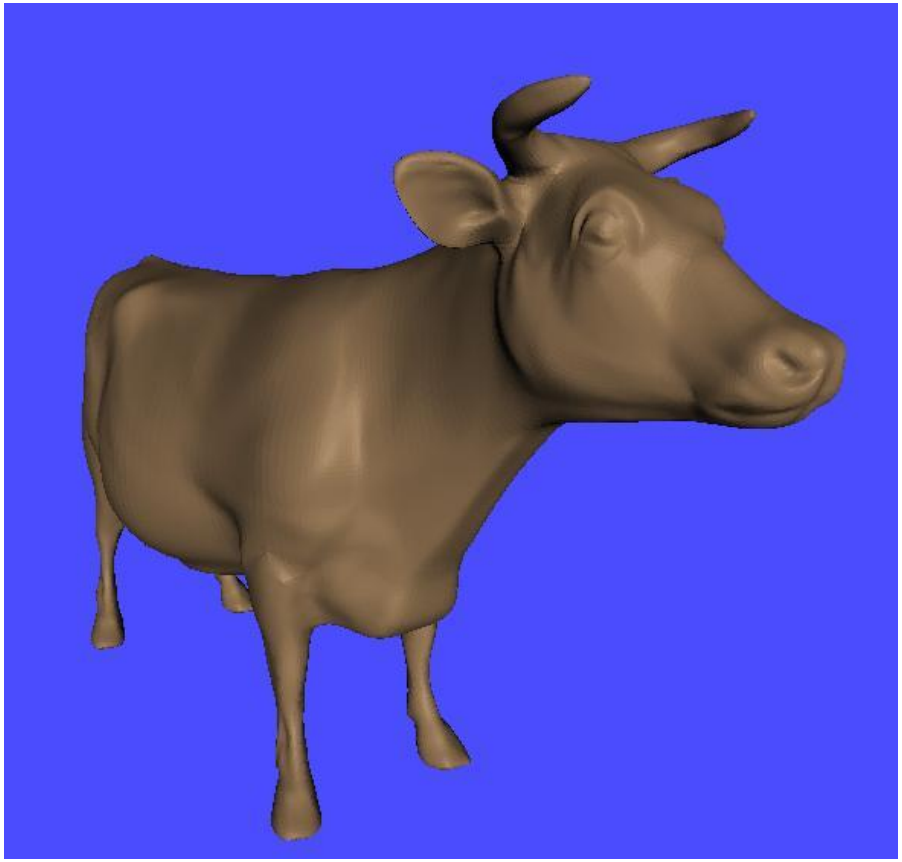
- The human visual system has a remarkable sensitivity to small differences in light intensity, due to a property known as **lateral inhibition**.
- If we see an increasing sequence of intensities, as is shown in the Figure, we perceive the increases in brightness as overshooting on one side of an intensity step and undershooting on the other, as shown in Figure below. We see stripes, known as **Mach bands**, along the edges.





Overcoming Flat Shading Limitations





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

Gouraud Surface Rendering

Gouraud surface shading was developed in the 1970s by Henri Gouraud

Worked at the University of Utah along with Ivan Sutherland and David Evans

Often also called **intensity-interpolation surface rendering**

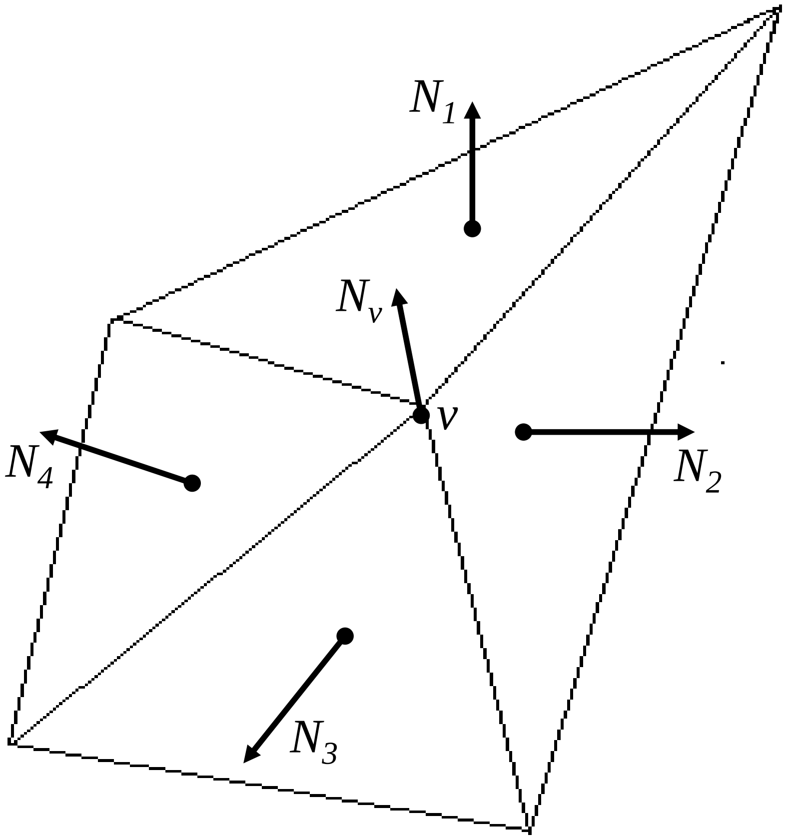
Intensity levels are calculated at each vertex and interpolated across the surface

Gouraud Surface Rendering (cont…)

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 Surface Rendering (cont…)



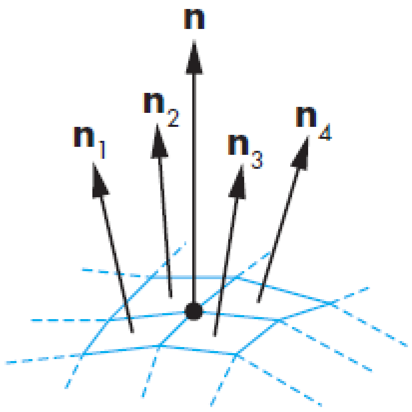
The average unit normal vector at *v* is given as:

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

or more generally:

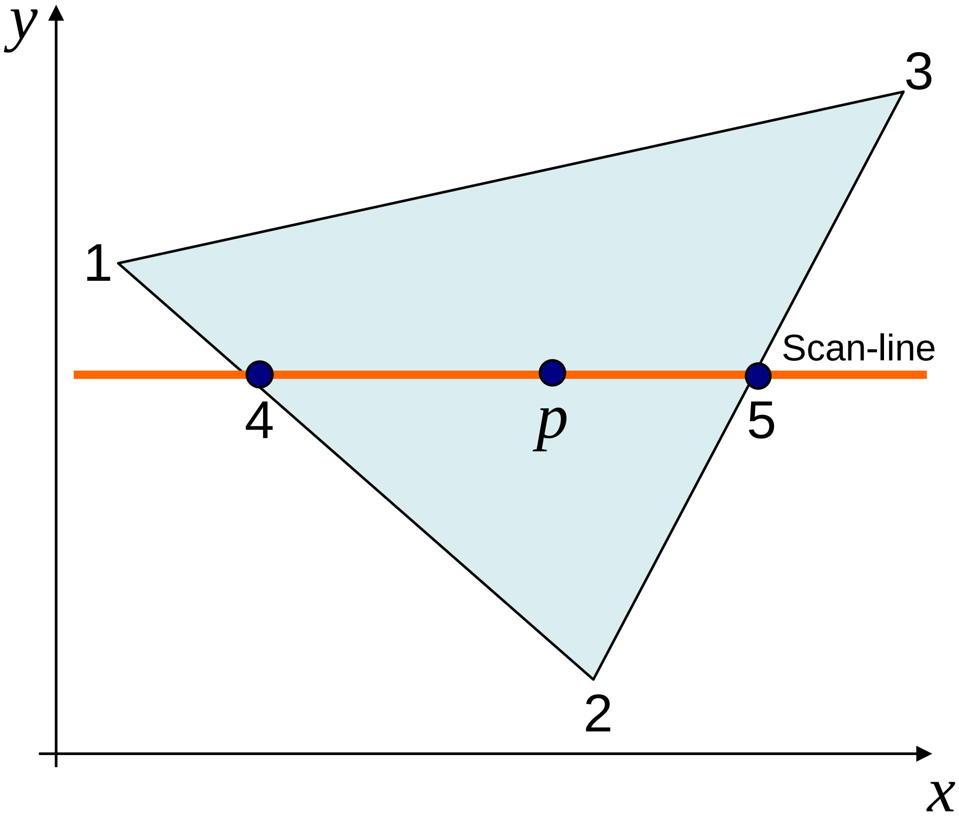
$$N_v = \frac{\sum_{k=1}^{n} N_k}{\left| \sum_{k=1}^{n} N_k \right|}$$

Normals near interior vertex



Gouraud Surface Rendering (cont…)

Illumination values are linearly interpolated across each scan-line

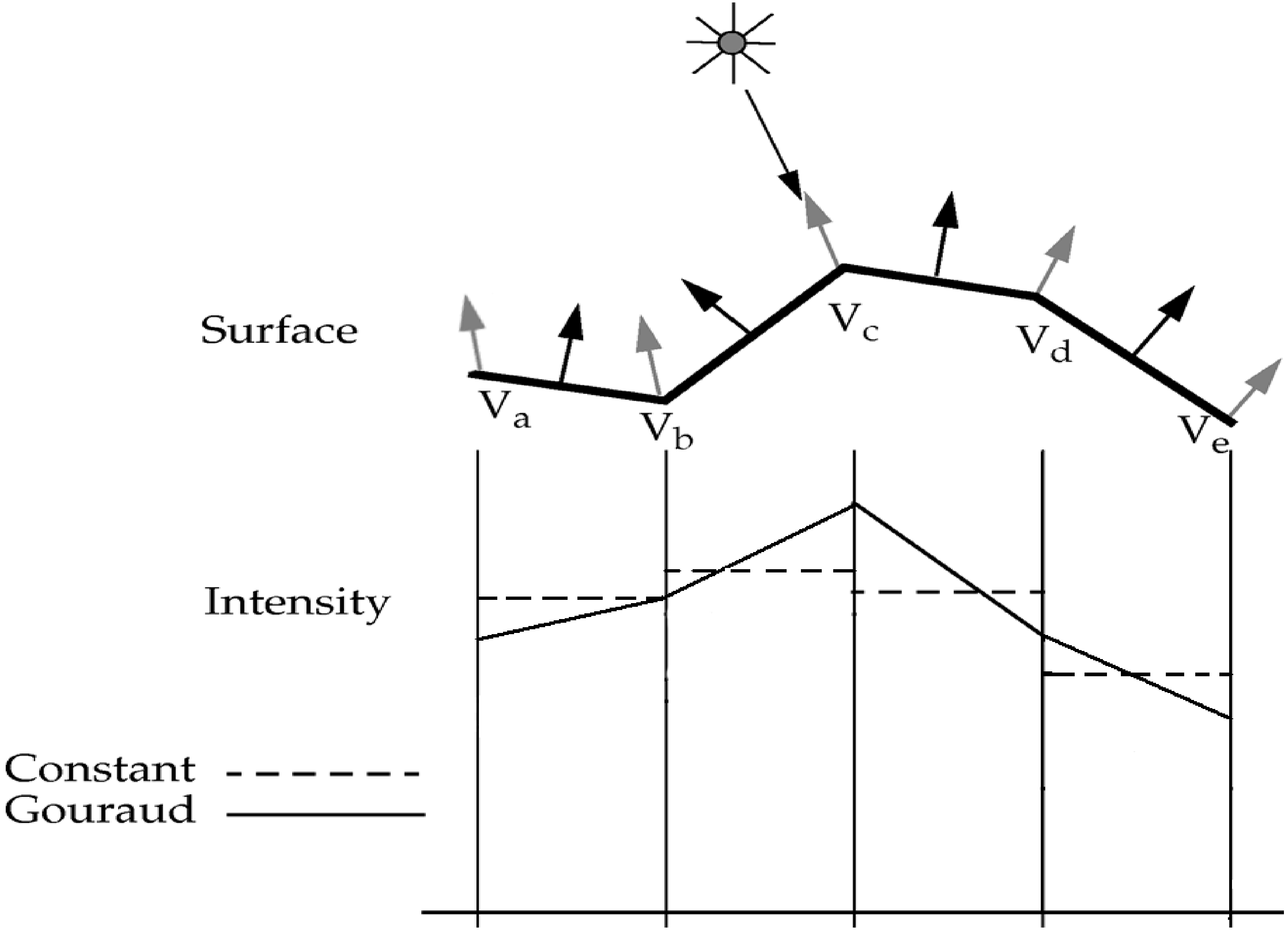


$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

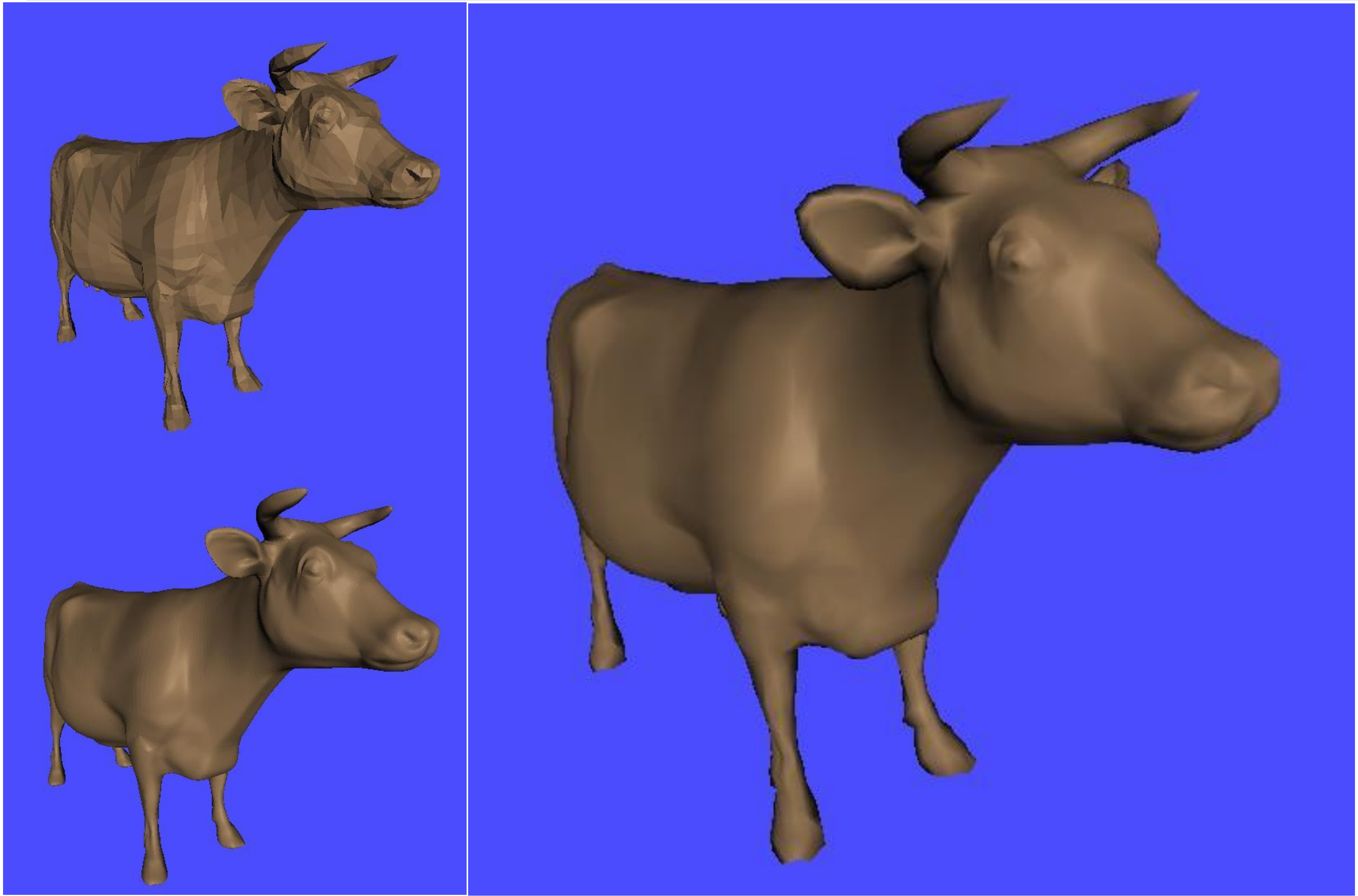
$$I_5 = \frac{y_5 - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_5}{y_3 - y_2} I_2$$

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

Advantages of Gouraud Surface Rendering



Gouraud Surface Rendering Example



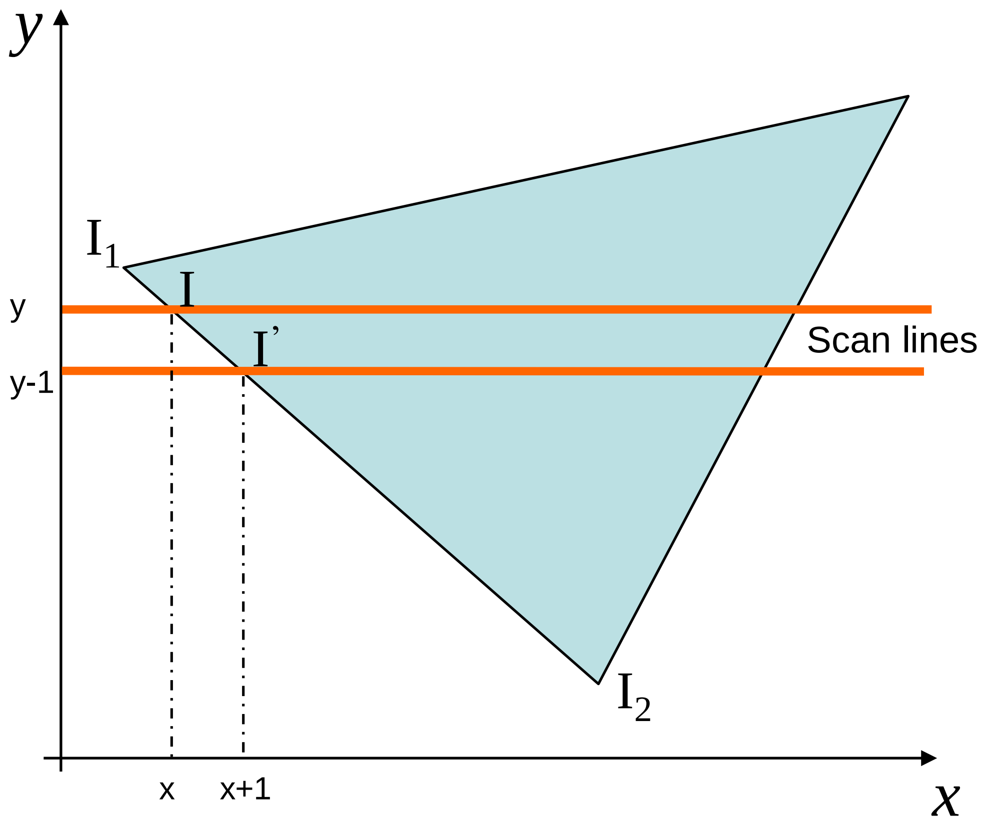
Gouraud Surface Rendering Implementation

Gouraud surfacing rendering can be implemented relatively efficiently using an iterative approach

Typically Grouaud shading is implemented as part of a visible surface detection technique

Gouraud Rendering Incremental calculation of intensities

Scan-line y is the next scan line below the vertex at y_1 with intensity I_1, that is $y = y_1 - 1$



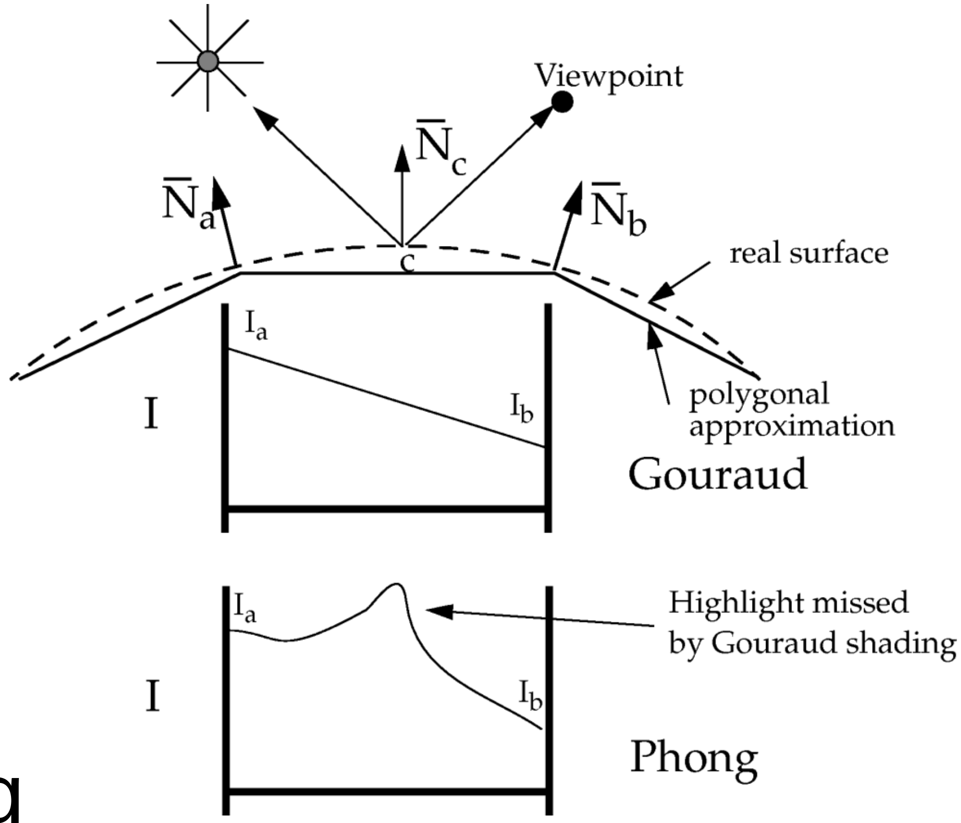
$$I = I_1 + \frac{I_2 - I_1}{y_1 - y_2}$$

$$I' = I + \frac{I_2 - I_1}{y_1 - y_2}$$

Problems With Gouraud Shading

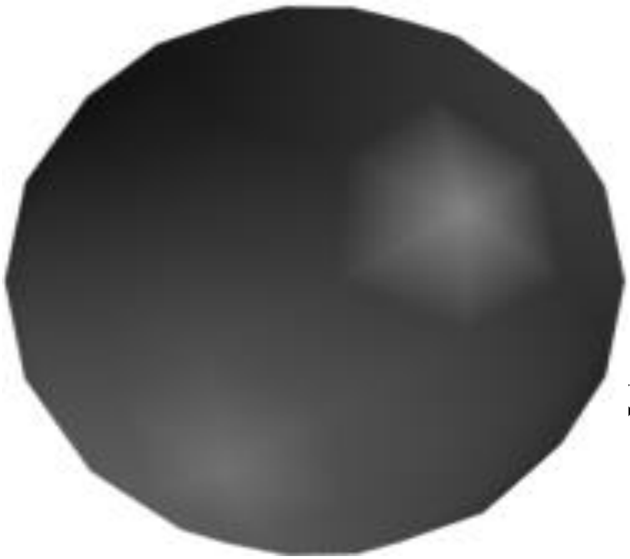
Gouraud shading tends to miss certain highlighting

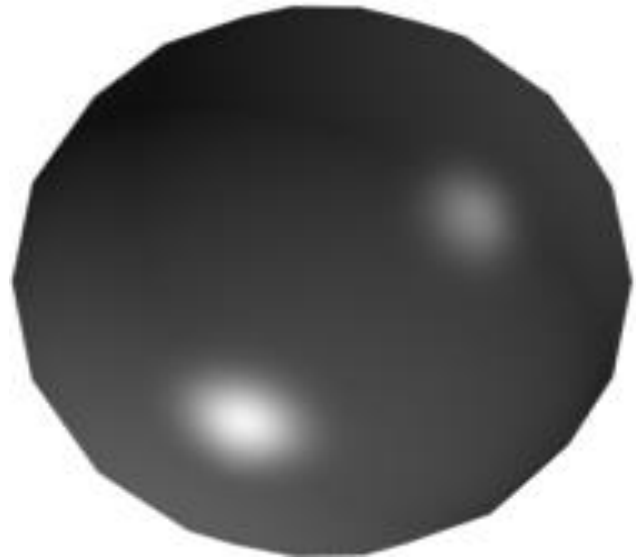
In particular Gouraud shading has a problem with specular reflections



Also, Gouraud shading can introduce anomalies known as **Mach bands**

Problems With Gouraud Shading (cont…)





The major problem with Gouraud shading is in handling specular reflections

Phong Surface Rendering

A more accurate interpolation based approach for rendering a polygon was developed by Phong Bui Tuong

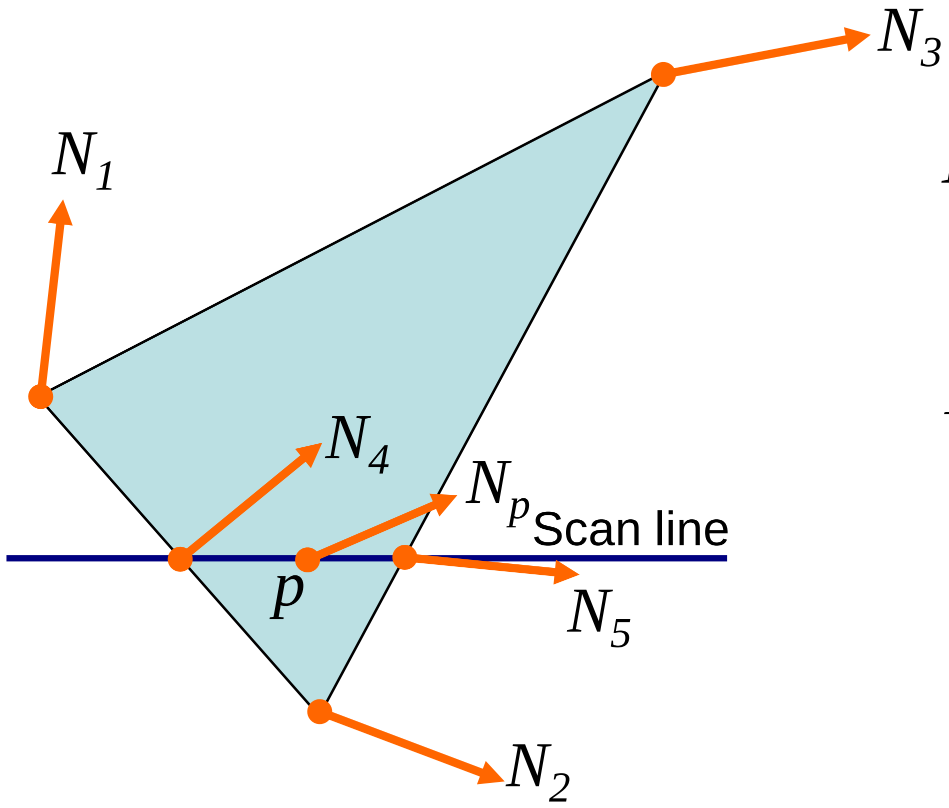
Basically the Phong surface rendering model (or **normal-vector interpolation rendering**) interpolates normal vectors instead of intensity values

Phong Surface Rendering (cont…)

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 Surface Rendering (cont…)



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

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

Phong Surface Rendering Implementation

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

Typically Phong shading is implemented as part of a visible surface detection technique

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