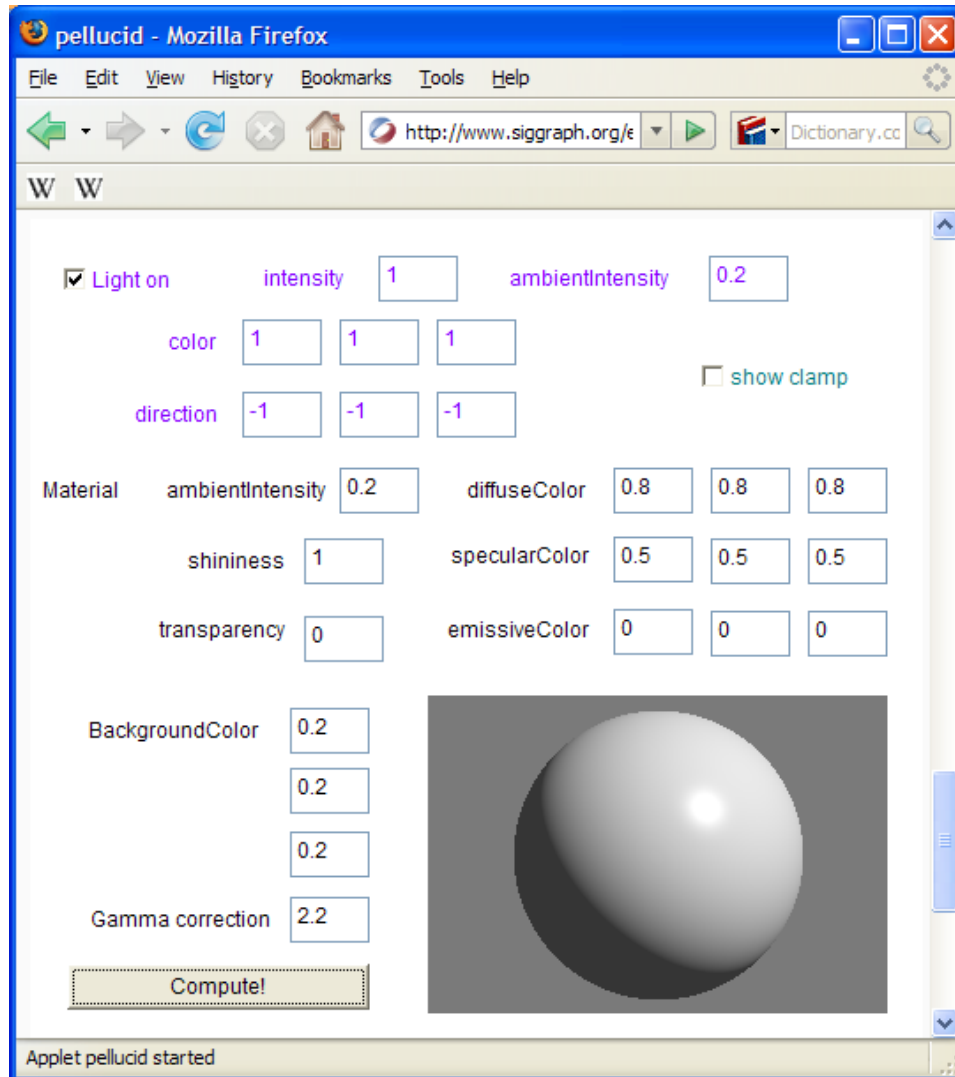


Interesting Illumination Demo



There's a very nice Java illumination model demo which may help you understand the effects of different kinds of reflections available at:

<http://www.siggraph.org/education/materials/HyperGraph/illumination/pellucid.html>

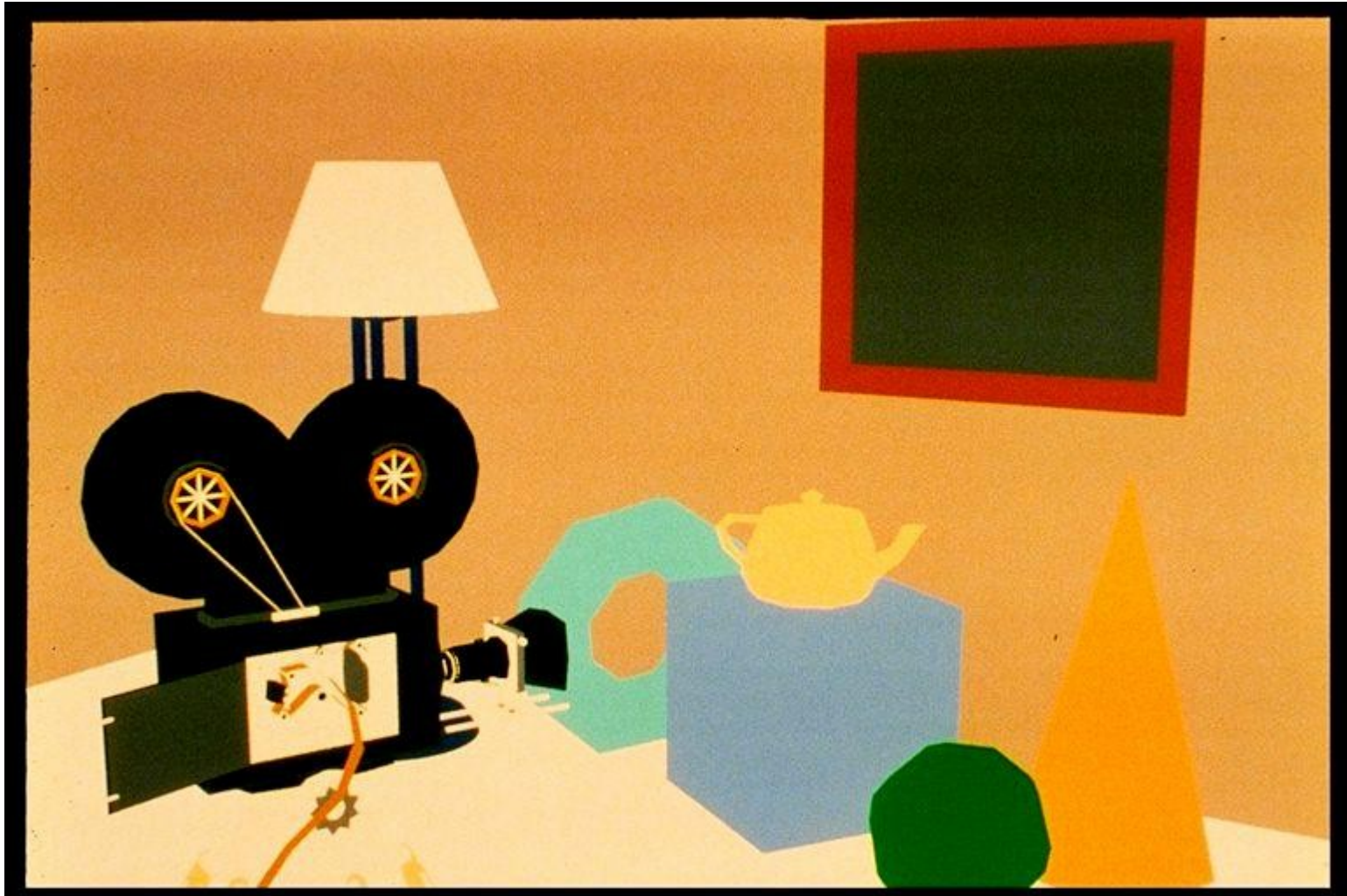
Try playing with the various parameters and see if you can predict what the sphere will look

Computer Graphics 16: Polygon Rendering Methods

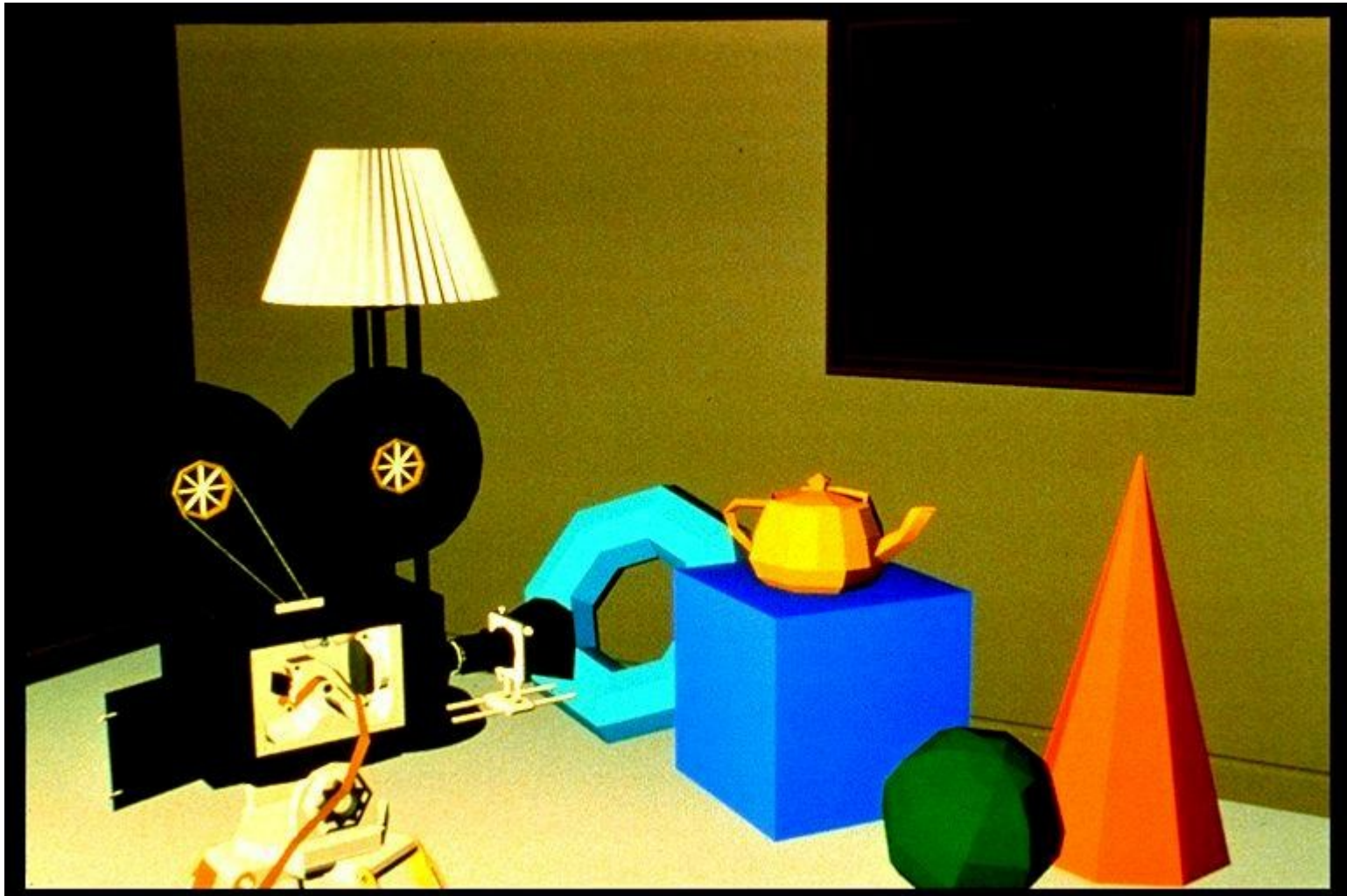
Today we will start to look at rendering methods used in computer graphics

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

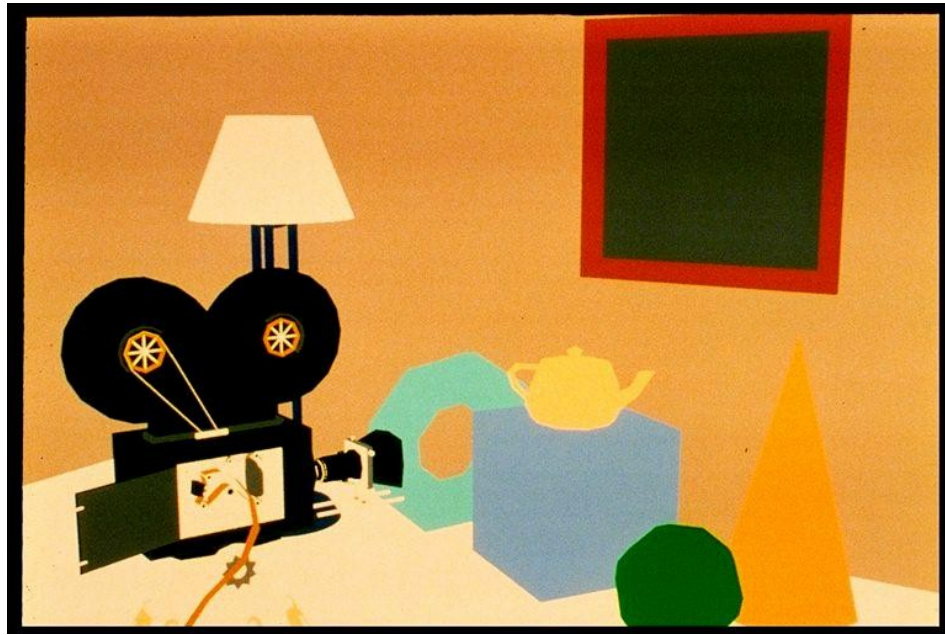
No Surface Rendering



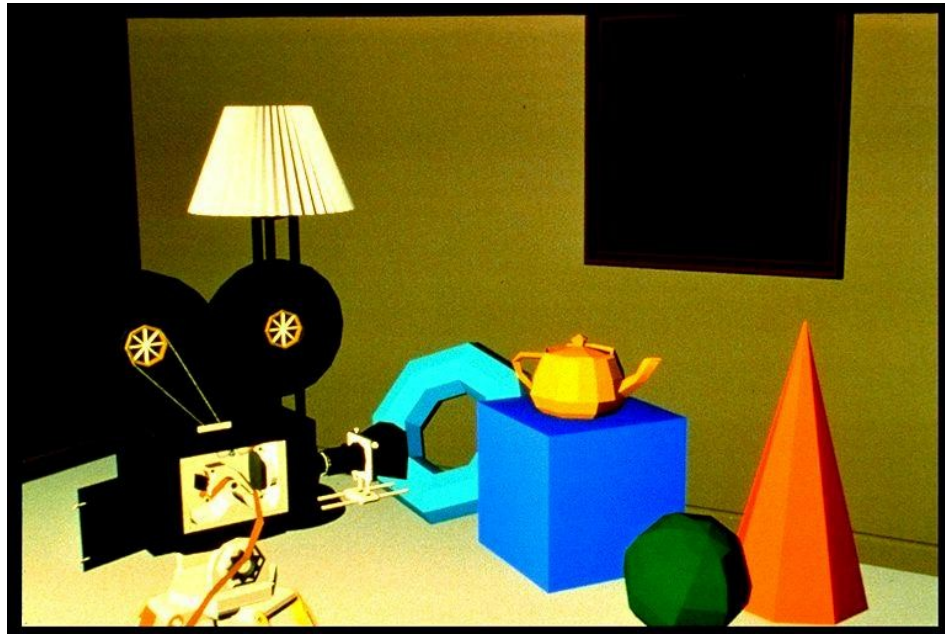
Flat Surface Rendering



No Surface Rendering Vs Flat Surface Rendering

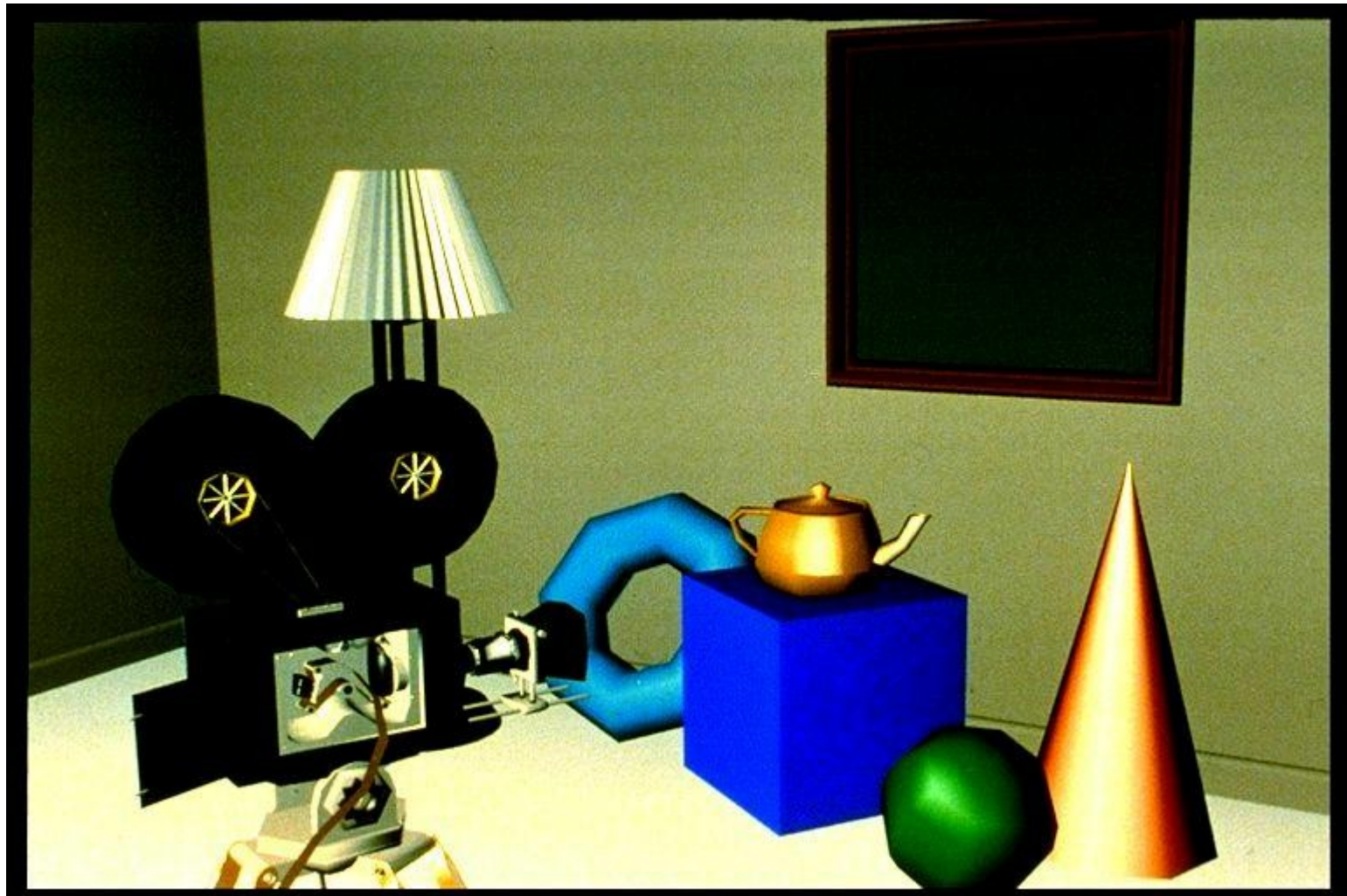


No Surface Rendering

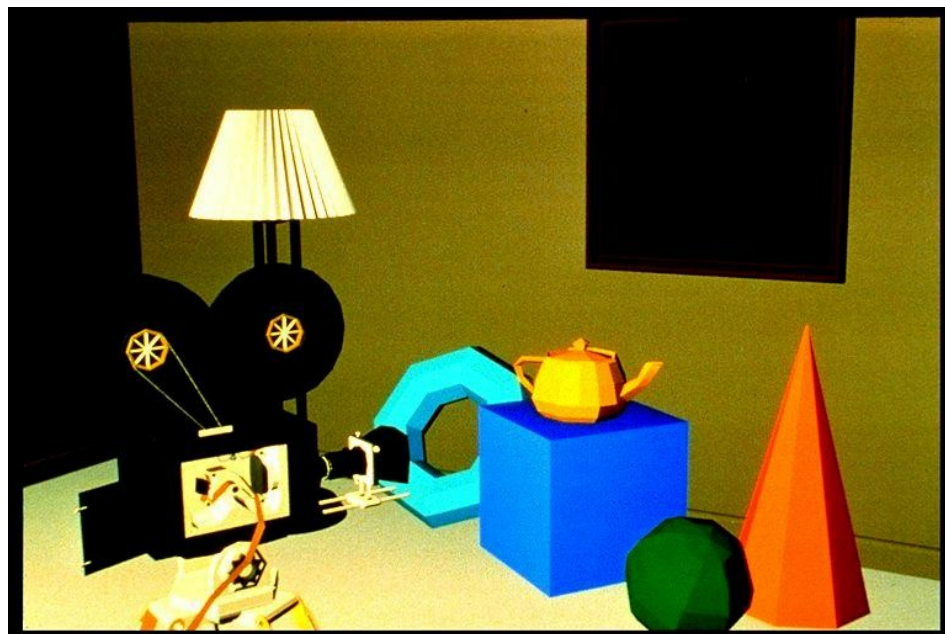


Flat Surface Rendering

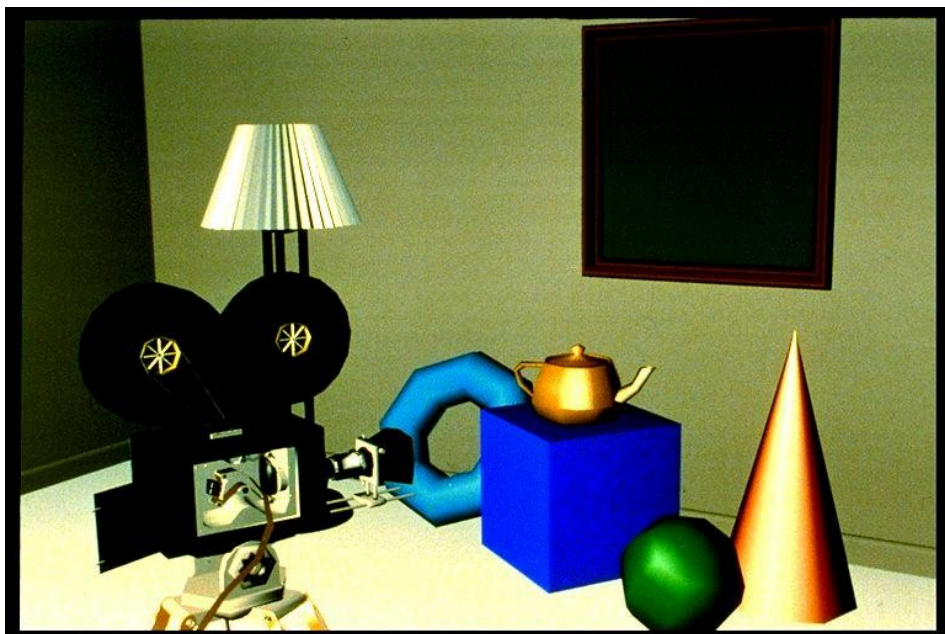
Gouraud Surface Rendering



No Surface Rendering Vs Flat Surface Rendering

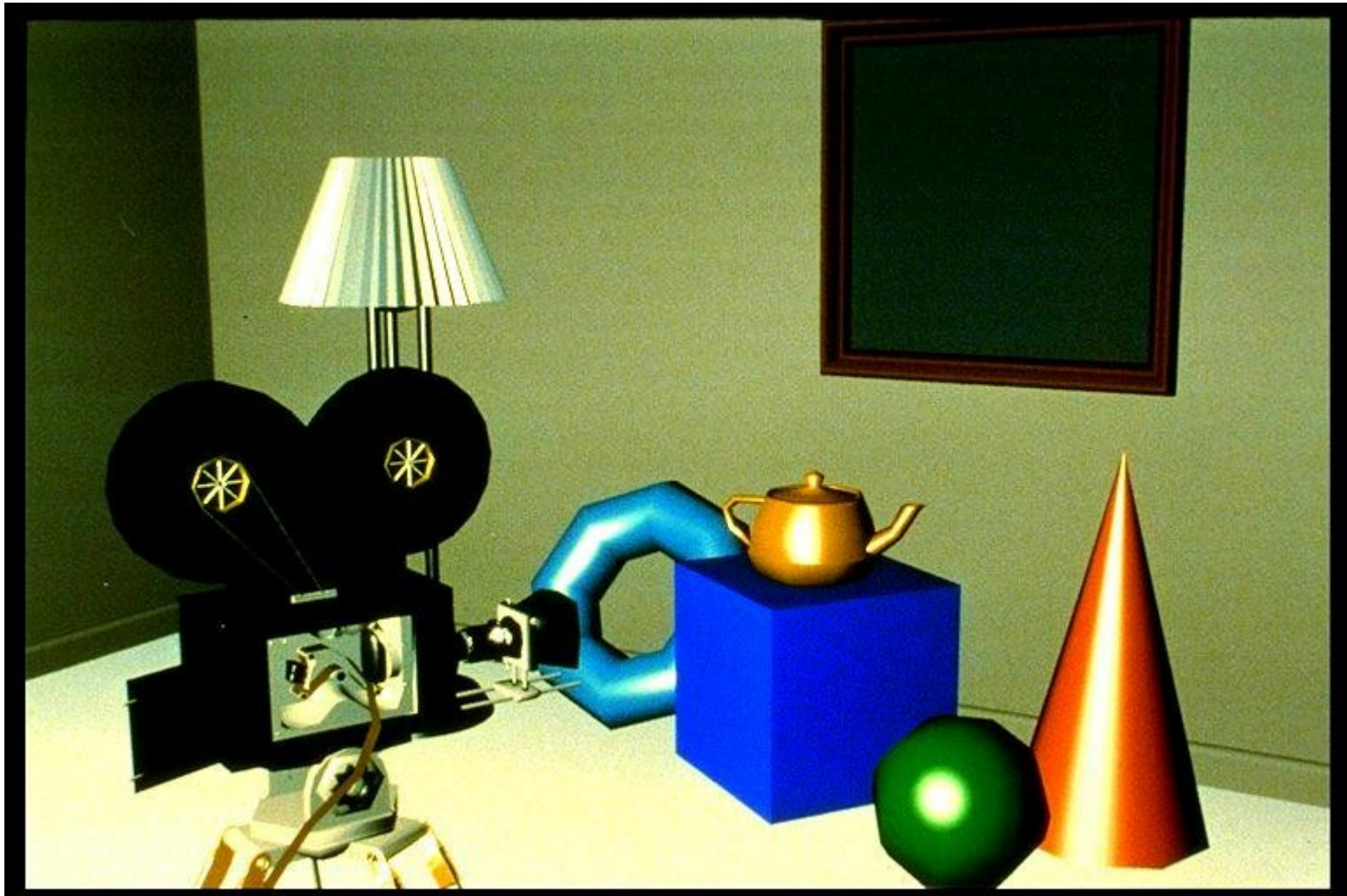


Flat Surface Rendering

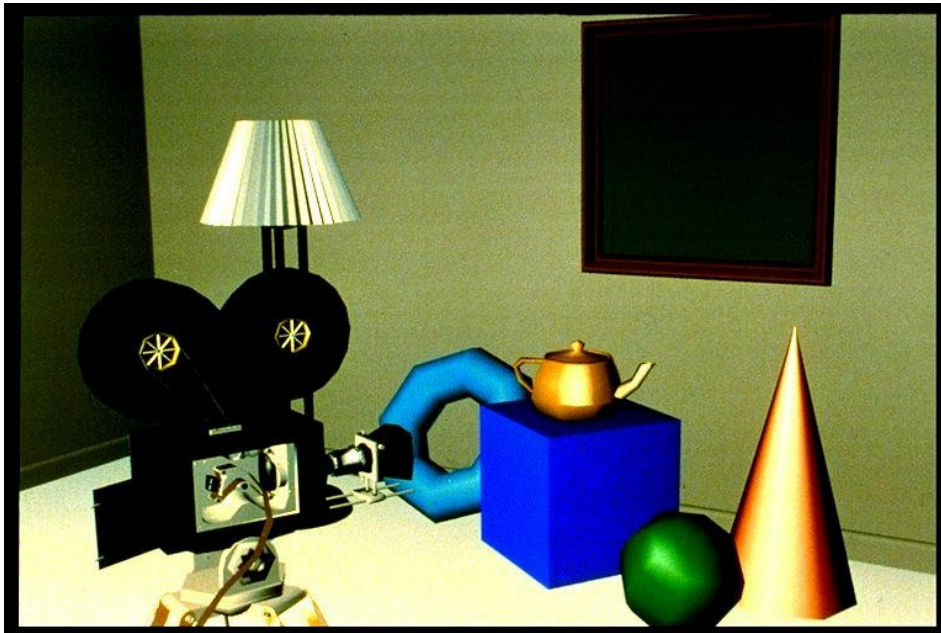


Gouraud Surface Rendering

Phong Surface Rendering



No Surface Rendering Vs Flat Surface Rendering



Gouraud Surface Rendering



Phong Surface Rendering

Flat Surface Rendering

The simplest method for rendering a polygon surface

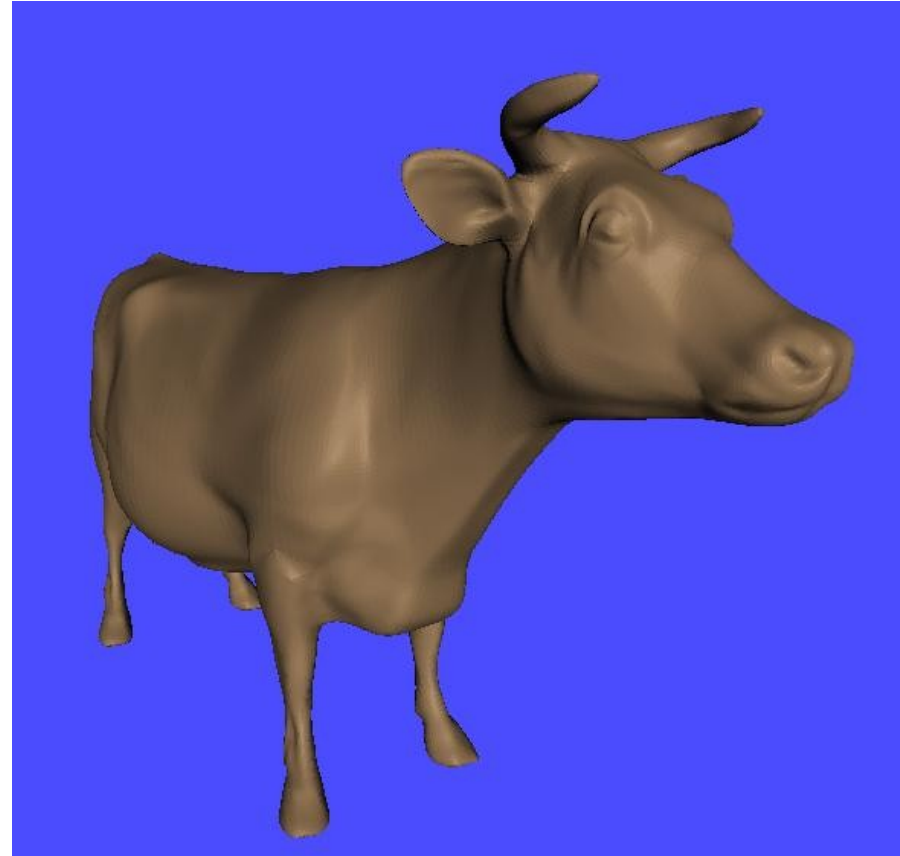
The same colour is assigned to all surface positions

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



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

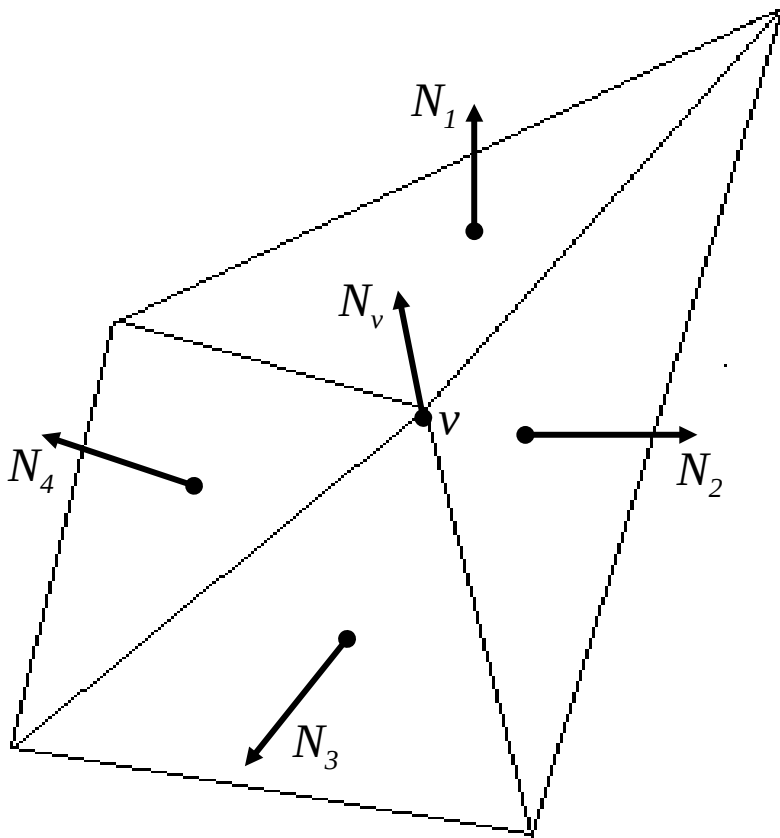
Gouraud Surface Rendering (cont...)

The average unit normal vector at v is given as:

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

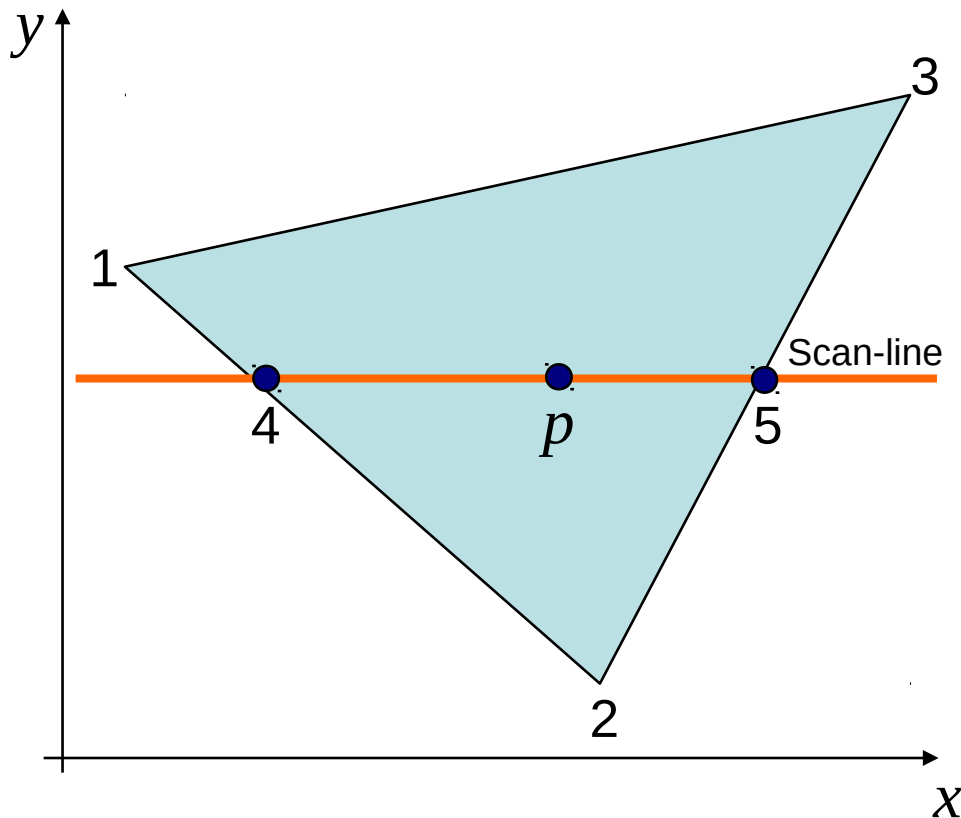
or more generally:

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



Gouraud Surface Rendering (cont...)

Illumination values are linearly interpolated across each scan-line

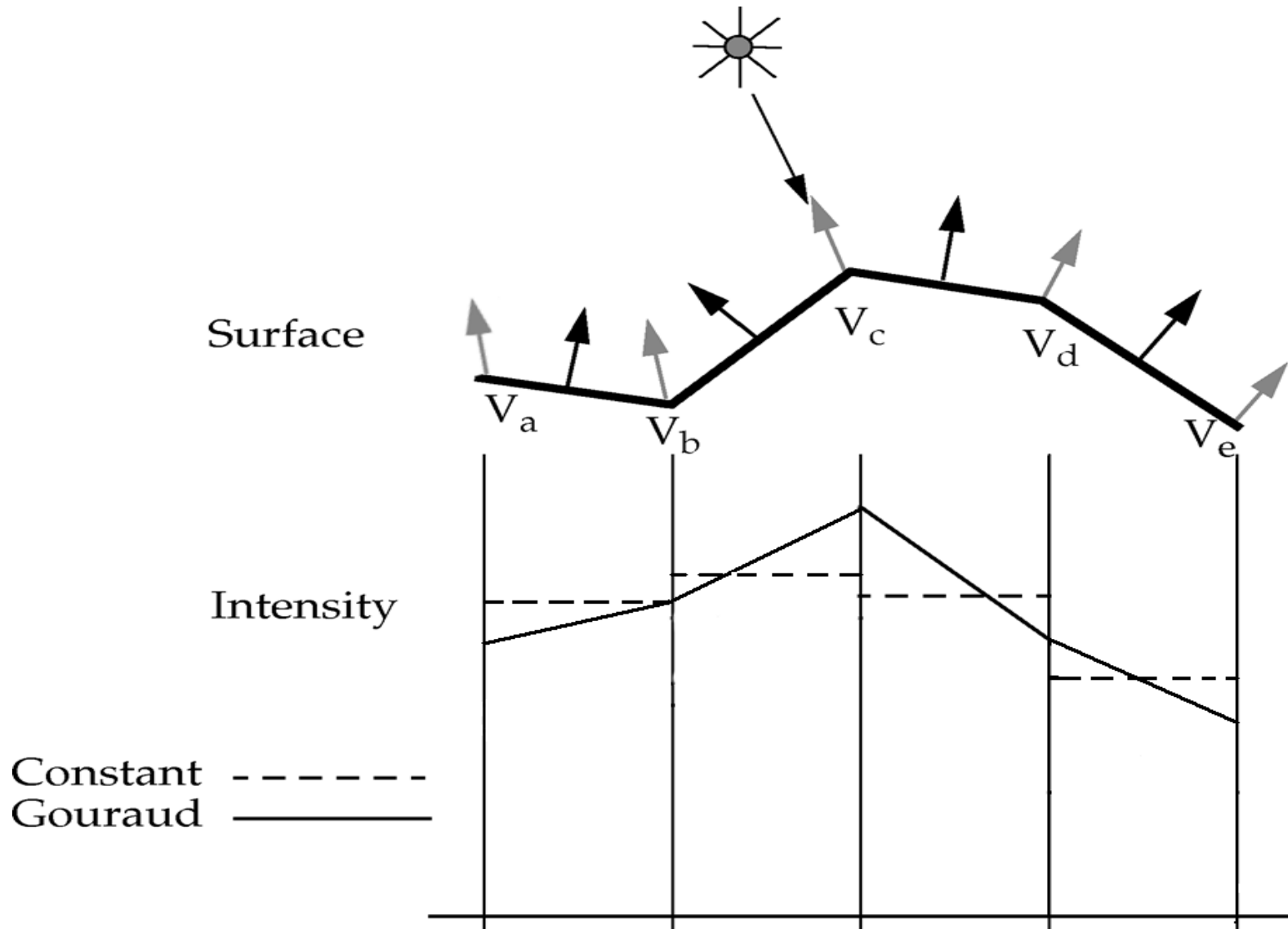


$$I_4 = \frac{y_4 - y_2}{y_1 - y_2} I_1 + \frac{y_1 - y_4}{y_1 - y_2} I_2$$

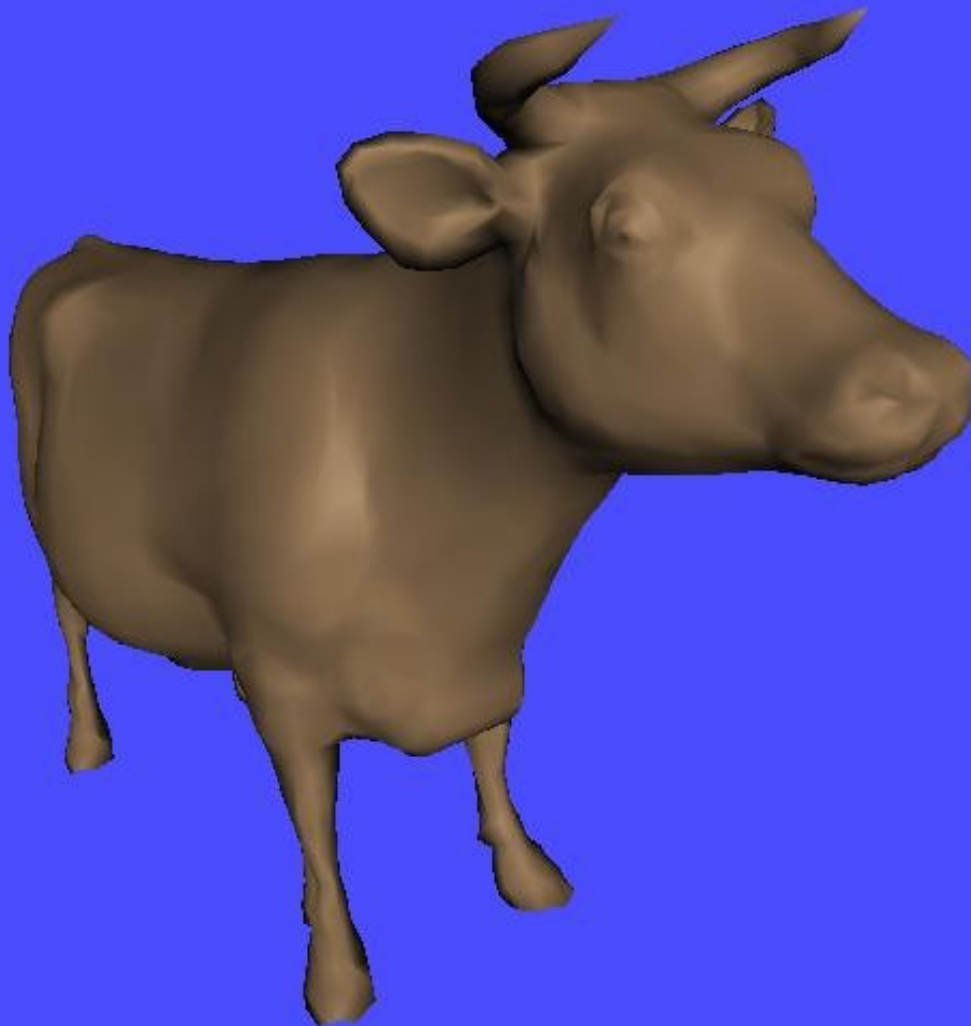
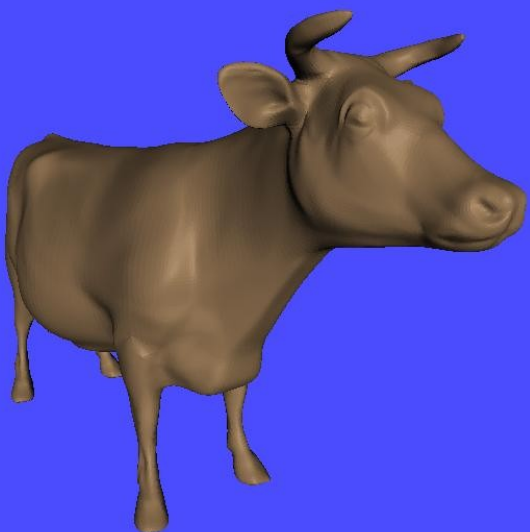
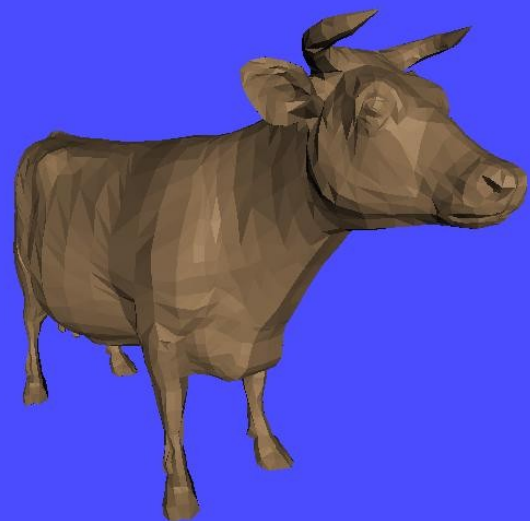
$$I_5 = \frac{y_5 - y_2}{y_3 - y_2} I_3 + \frac{y_3 - y_5}{y_3 - y_2} I_2$$

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

Advantages of Gouraud Surface Rendering



Gouraud Surface Rendering Example



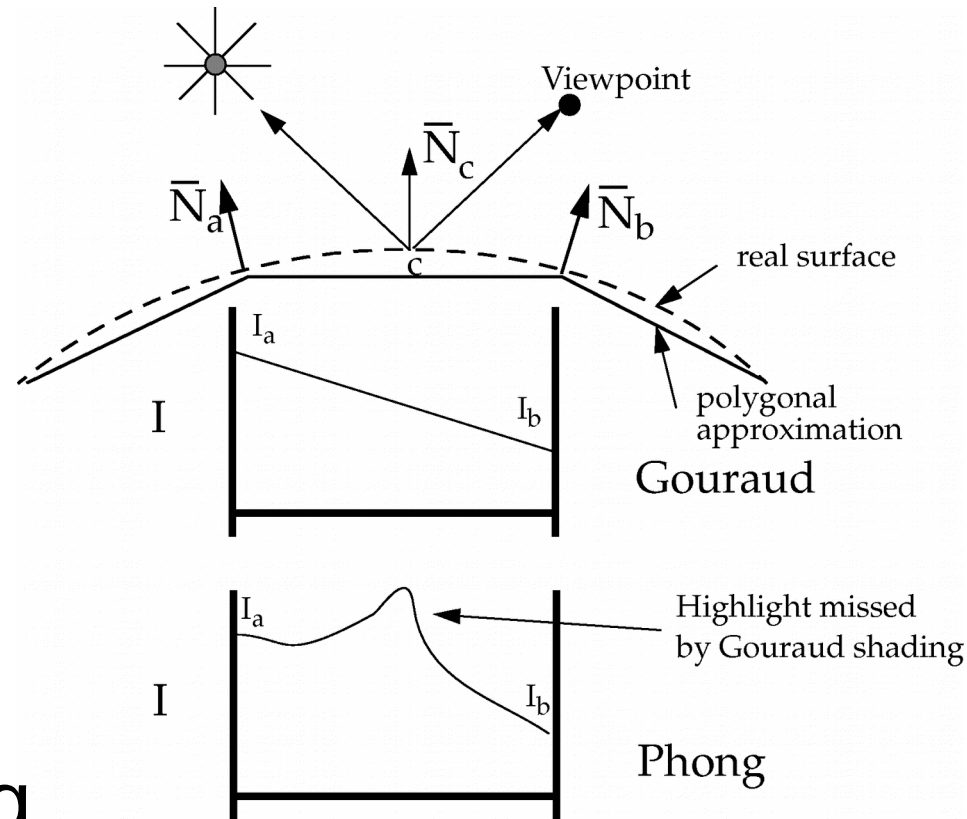
Gouraud surfacing rendering can be implemented relatively efficiently using an iterative approach

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

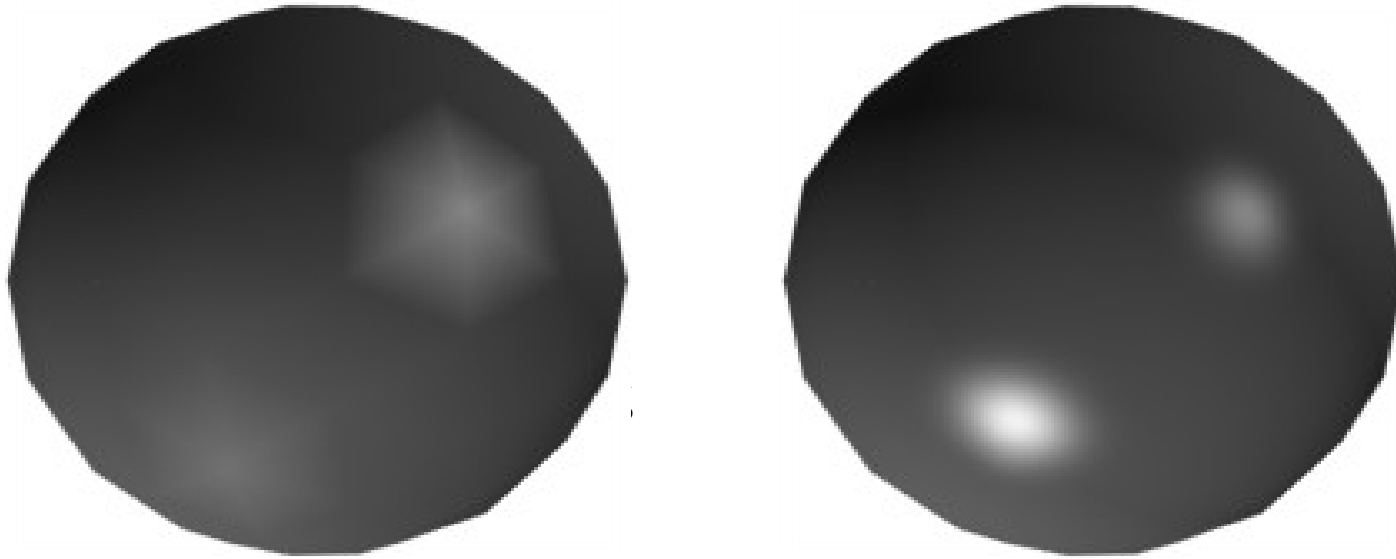
Problems With Gouraud Shading

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

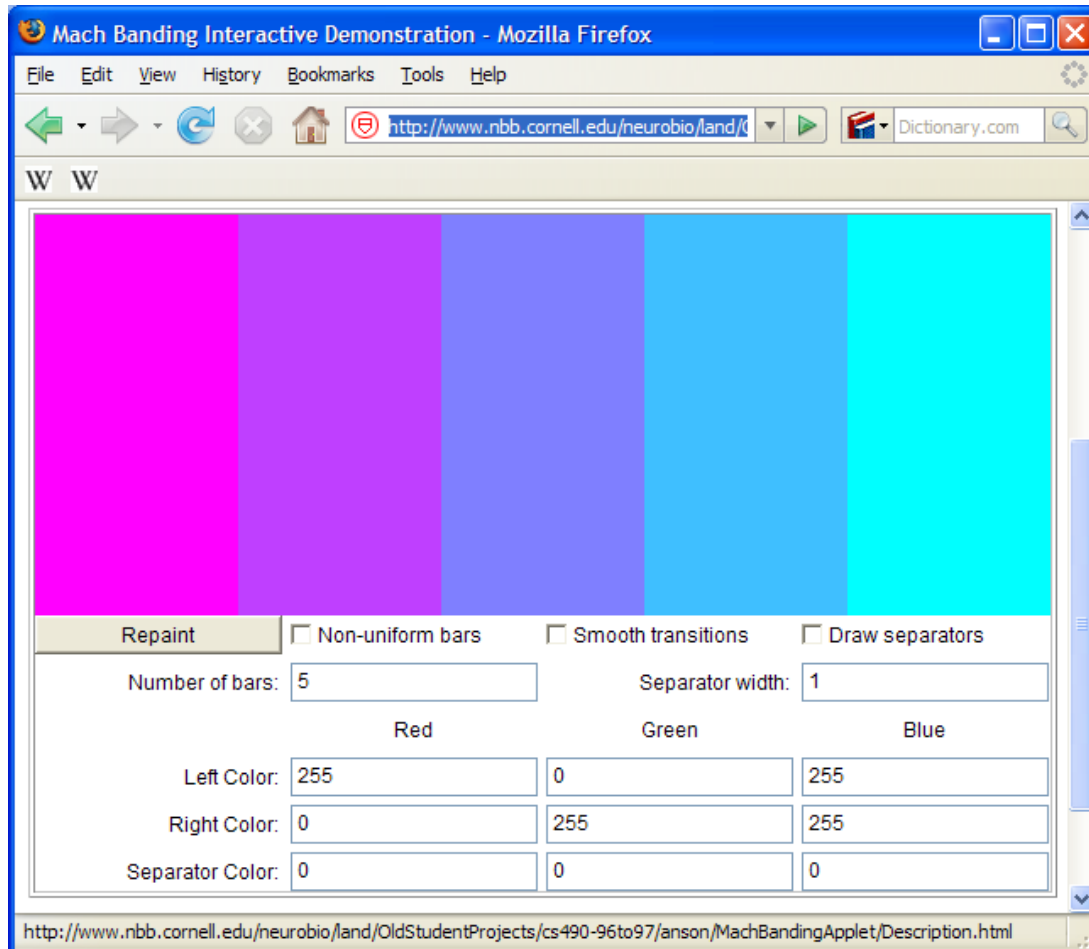
Mach Bands

A psychological phenomenon whereby we see bright bands where two blocks of solid colour meet

A good demo is available to experiment with this at:

<http://www.nbb.cornell.edu/neurobio/land/OldStudentProjects/cs490-96to97/anson/MachBandingApplet/>

Try playing with the various parameters and see if you can predict what the sphere will look like



Phong Surface Rendering

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

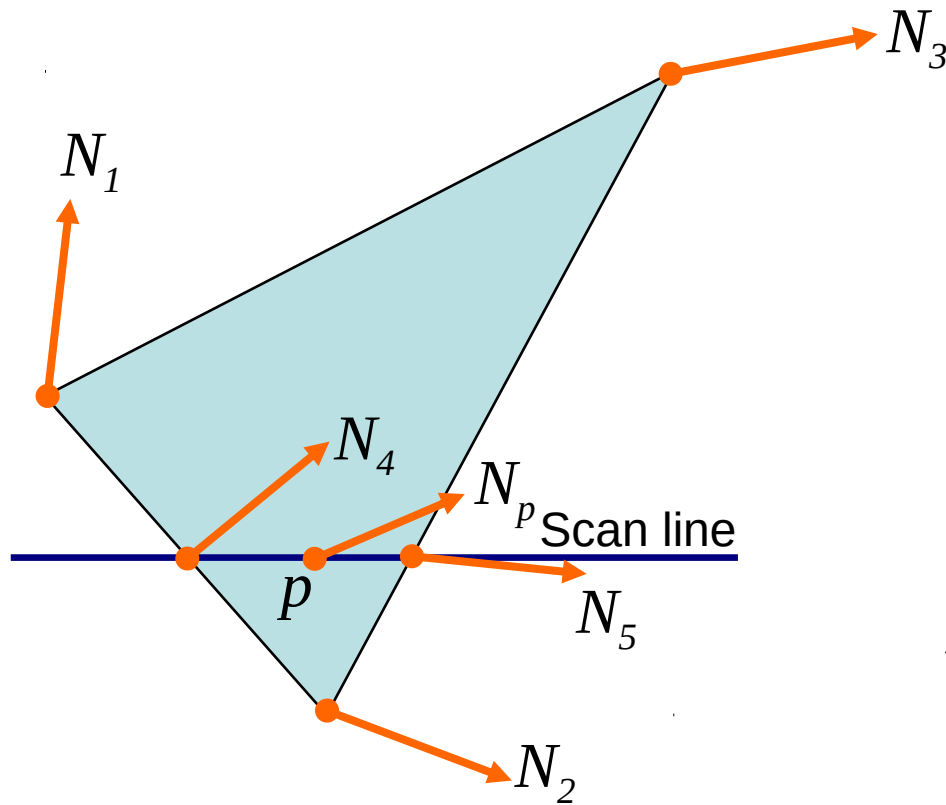
Basically the Phong surface rendering model (or **normal-vector interpolation rendering**) interpolates normal vectors instead of intensity values

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...)



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

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

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

Phong shading is much slower than Gouraud shading as the lighting model is reevaluated 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



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