Shading &
Smooth Shading

Graphics Systems /

Computer Graphics and Interfaces

Shading & Smooth Shading

Objective: calculating the color of each point of the visible surfaces.

Solution ***brute-force***: Calculate the normal at each point and apply the desired illumination model.

Different methods:

1. Constant shading
2. Interpolated shading = *Smooth Shading*
    1. Gouraud method
    2. Phong Method

Shading

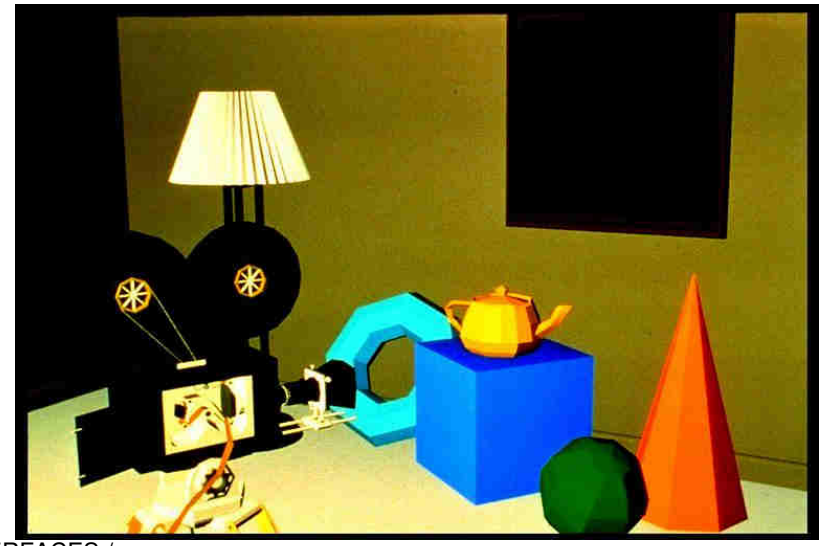
Constant shading

The color is calculated only for one point of the polygon and replicated in all other points of the same polygon.

This method is equivalent to the following conditions:

- The light source is at infinity, so that NL is constant at any point of the polygon (parallel rays).
- The observer is at infinity so that RV is constant at any point of the polygon
- The face is the flat surface of the model itself and is not an approximation of a curved surface

The polygonal mesh is noticeable
Mach Band Effect, with discontinuity of the light function



Shading

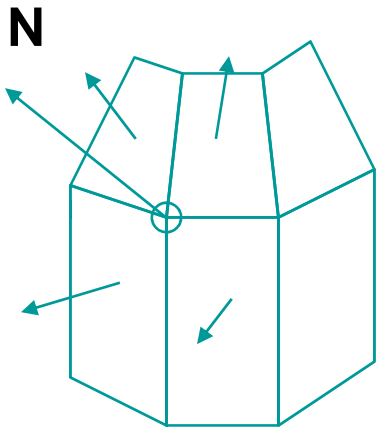
Interpolated shading or *Smooth Shading*

In the previous solution, when approaching a curved surface by a polygonal mesh, we found discontinuity in color between adjacent polygons (Mach Band effect, with discontinuity of illumination function).

The following solutions will surpass this problem by determining the color of a point based on a interpolation from the vertices of the polygon.

Required: normals, on the vertices, to the original surface

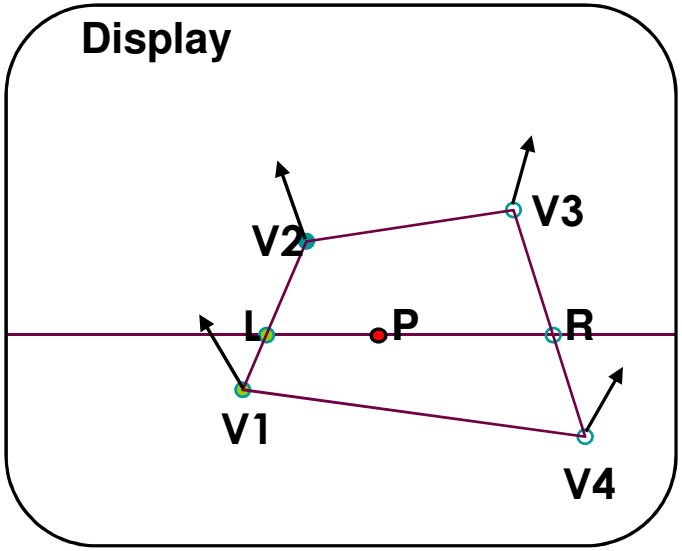
1. Analytical solution:
   - Analytical expression of the surface ...
2. Approximate solution:
   - Interpolation of the normals of neighboring polygons.



Smooth Shading

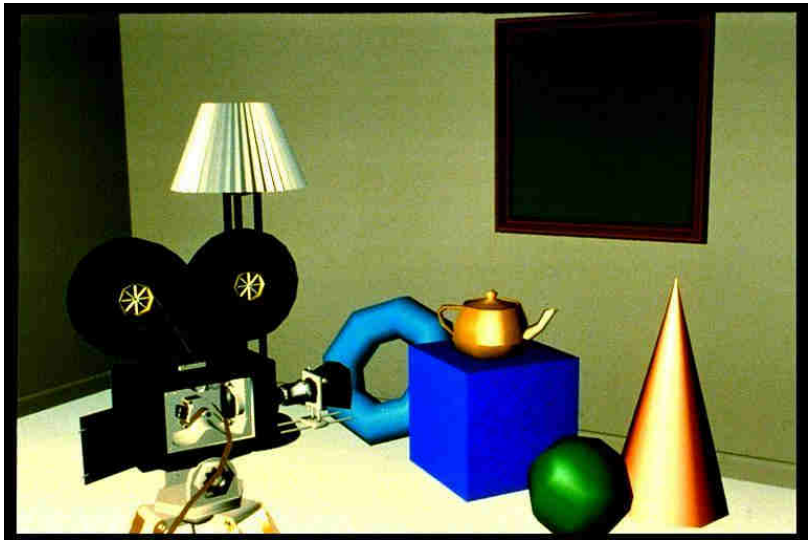
Gouraud Method

1. Calculate the color of each vertex using the desired illumination model.
2. Calculate the color of the remaining points of the polygon by bi-linear interpolation.



1. Color of point **L** is obtained by interpolating color in **V1** and **V2**
2. **R =** interpolation of **V3** and **V4**
3. **P =** interpolation of **L** and **R**

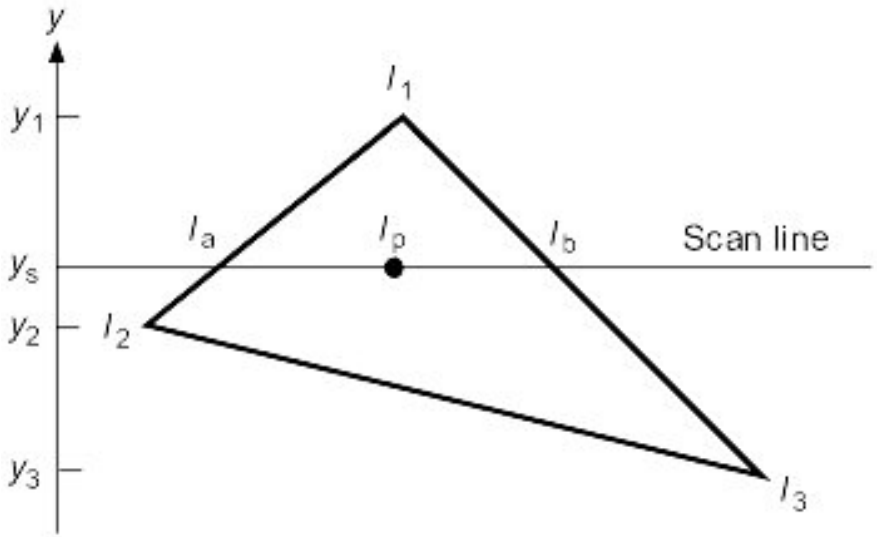
Location of edges are noticeable
Mach Band effect, with discontinuity of the derivative of the light function



Smooth Shading

Gouraud Method

Calculation of interpolated values



$$I_a = I_1 - (I_1 - I_2)\frac{y_1 - y_s}{y_1 - y_2}$$

$$I_b = I_1 - (I_1 - I_3)\frac{y_1 - y_s}{y_1 - y_3}$$

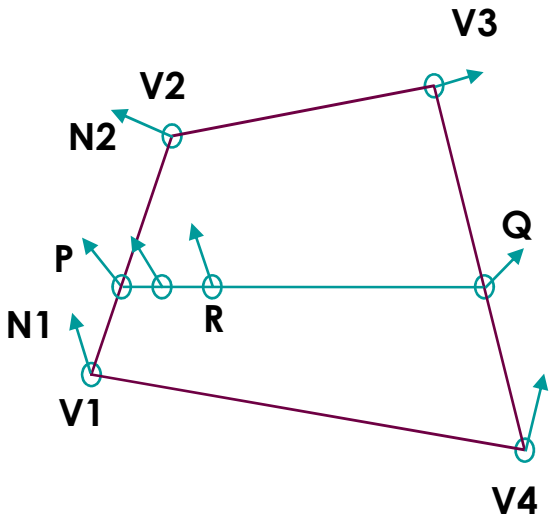
$$I_p = I_b - (I_b - I_a)\frac{x_b - x_p}{x_b - x_a}$$

Smooth Shading

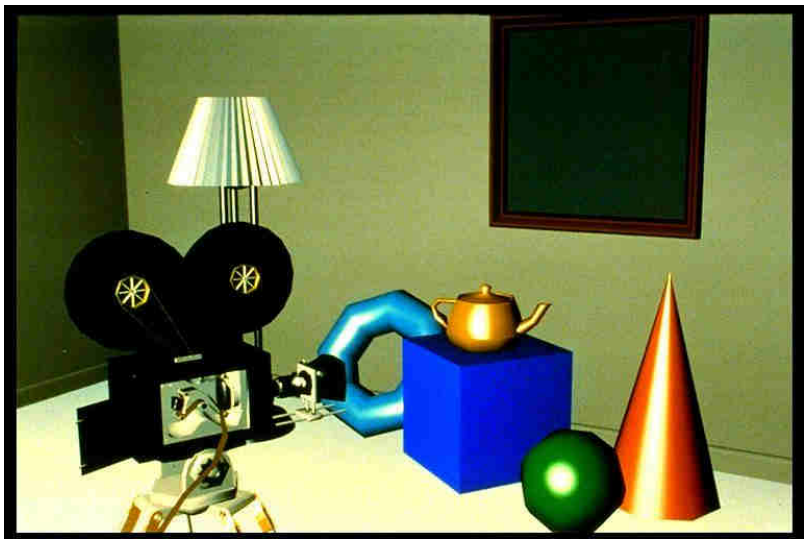
Phong Method

Performs **normals** interpolation instead of the color.

1. For each vertex of the polygonal mesh calculates the surface normal vector (by analytical expression or approximated by interpolation).
2. Normall in edges points are calculated by linear interpolation of the vertices normals. The normals in points along a scan line are obtained by linear interpolation of the normals in edges.
3. The local illumination model is used at each point.

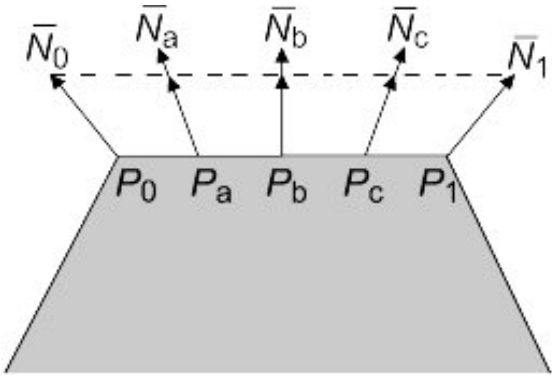


1. Normal in **P** obtained by interpolation of normals in **V1** and **V2.**
2. Normal in **Q =**interpolation of **V3** and **V4**
3. Normal in **R =** interpolation of normals in **P**and **Q**



***High-lights* well placed**
Mach Band Effect is not apparent

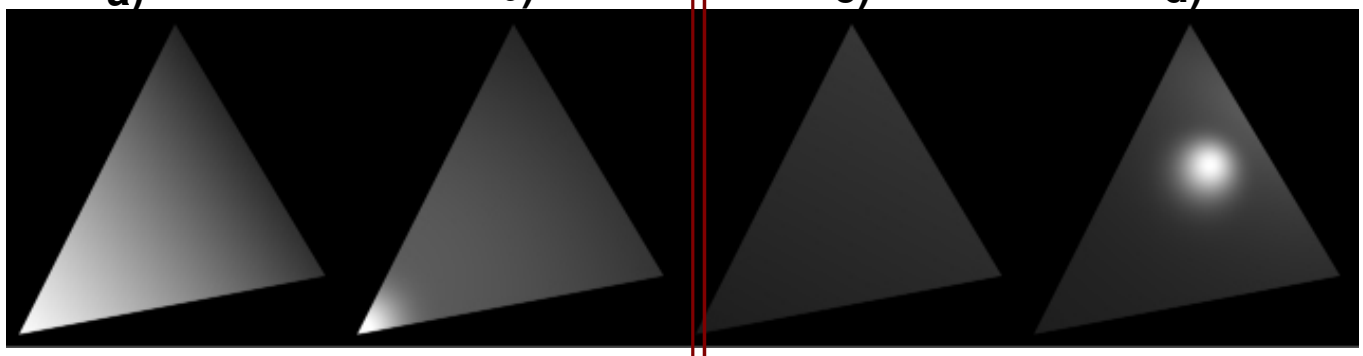
Smooth Shading



The lighting calculation on each *pixel* requires the inverse mapping for the object coordinates determined after normal;

Coord. Obj = F (Screen Coordinates)

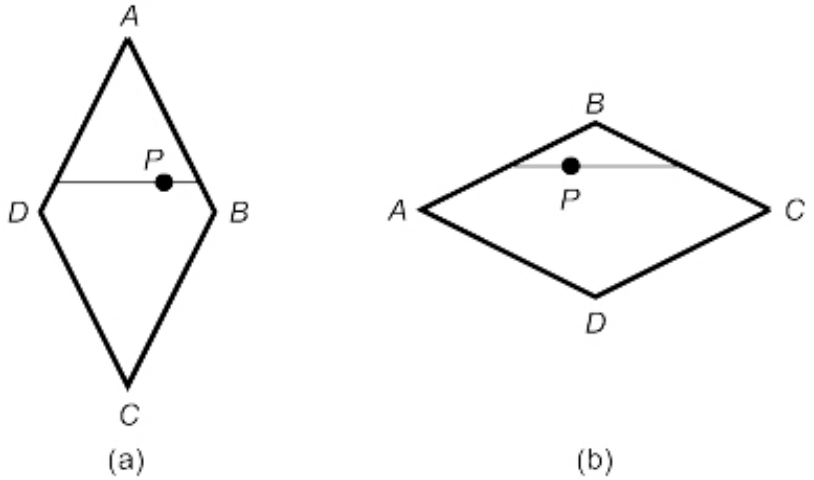
Specular reflection with Gouraud shading: a), c); With Phong: b) and d)



Maximum reflection in the left corner.

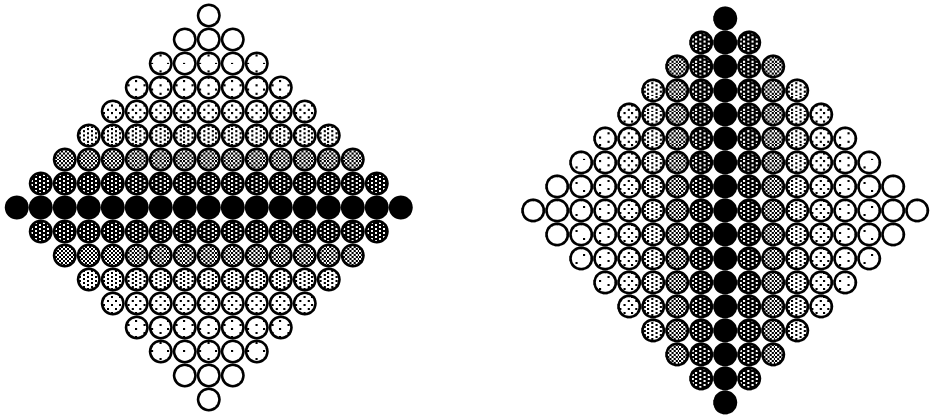
Maximum reflection in the center of the polygon.

Interpolated Shading Problem

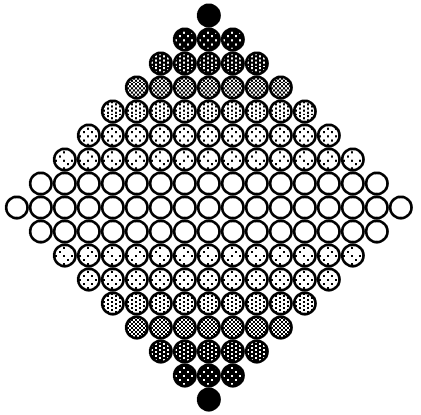


The result depends on the orientation of the polygon:

- In (a) the calculation of P uses the colors of the vertices A, D, B.
- In (b) the calculation of P uses the colors of the vertices A, B, C.



90 º rotation



Result

Textures

Graphics Systems /

Computer Graphics and Interfaces

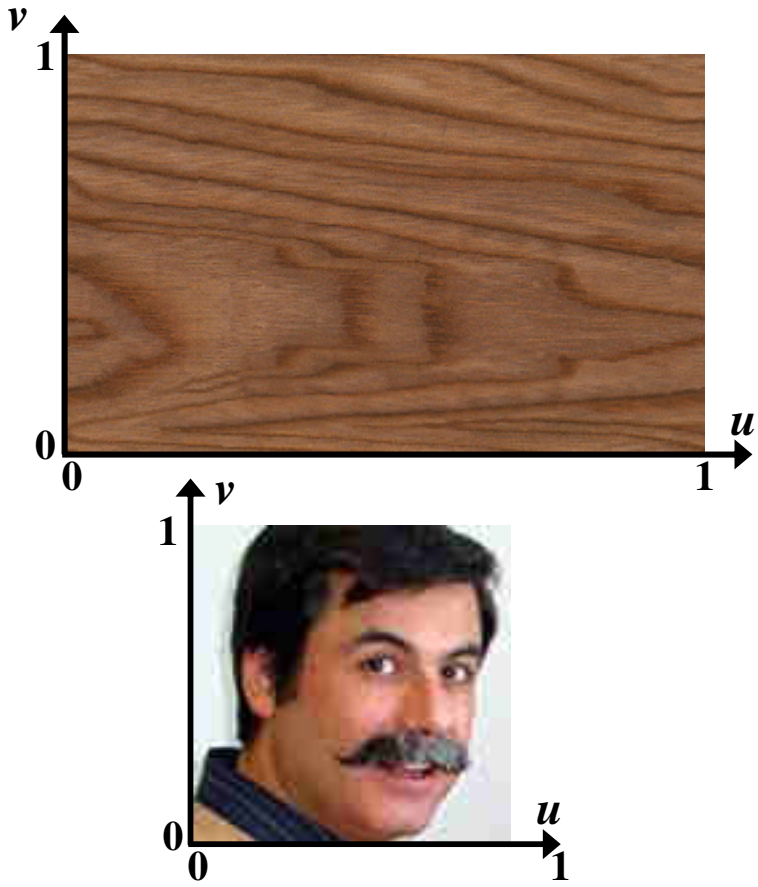
Textures

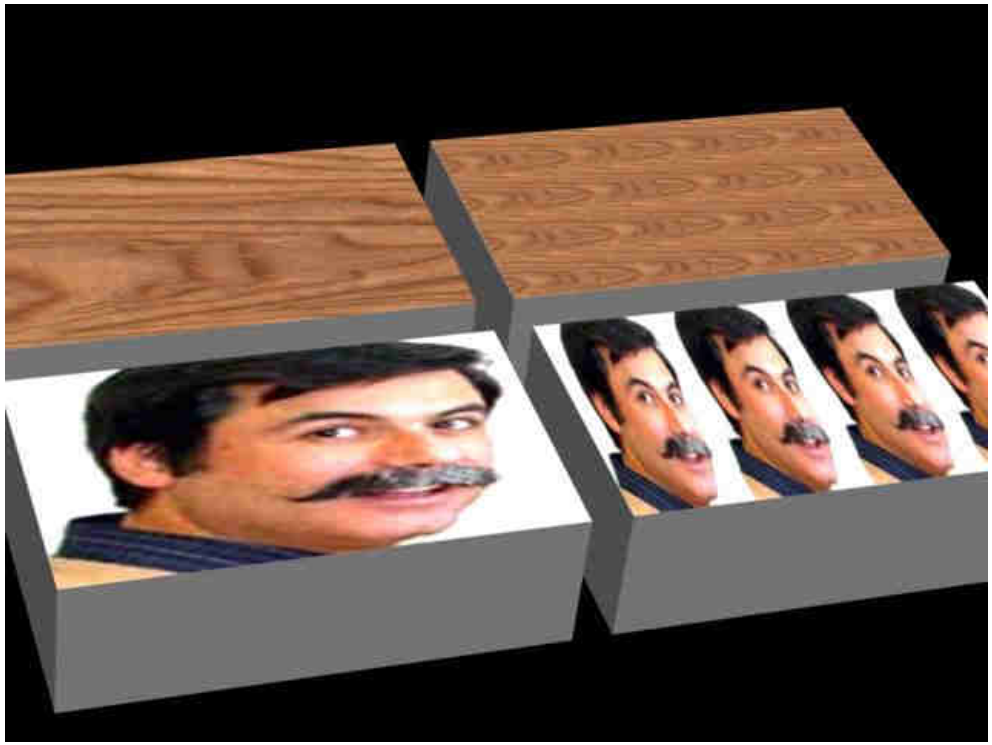
- Add visual detail without increasing the geometric detail

- Most common types
  - Texture mapping (2D images)
    - An image over a polygon (wallpaper)
      - Representation of a painting in a frame
      - Simulation of a landscape out of a window
      - Wooden surface
      - Etc. ...
  - *Bump Mapping Textures*
    - Besides the 2D image, it creates sense of relief (roughness)
      - Orange Peel
      - Strawberry Peel
      - Bricks
      - Etc. ...
  - 3D Textures
    - The texture evolves continuously "inside" the objects
      - Volume of Wood
      - Volume of Marble
      - Etc. ...

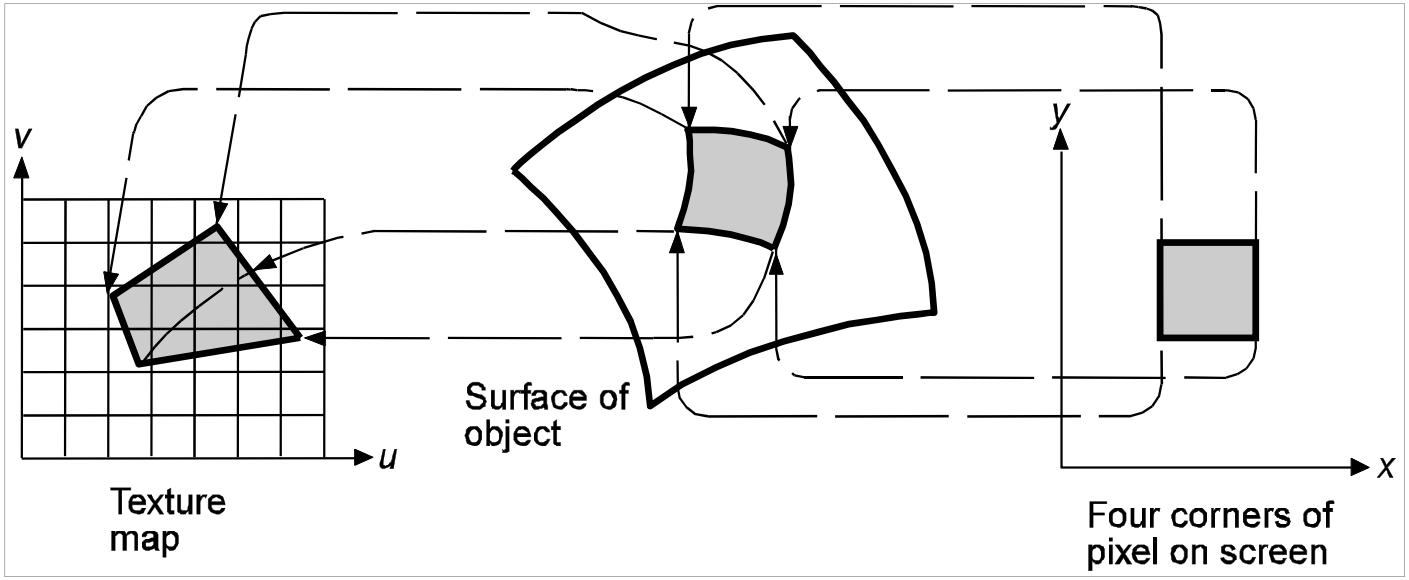
Texture Mapping (2D)

- Texture has standard coordinates $(u, v) \in [0,1]$
  - *Pixels* of the texture image are called *texels*





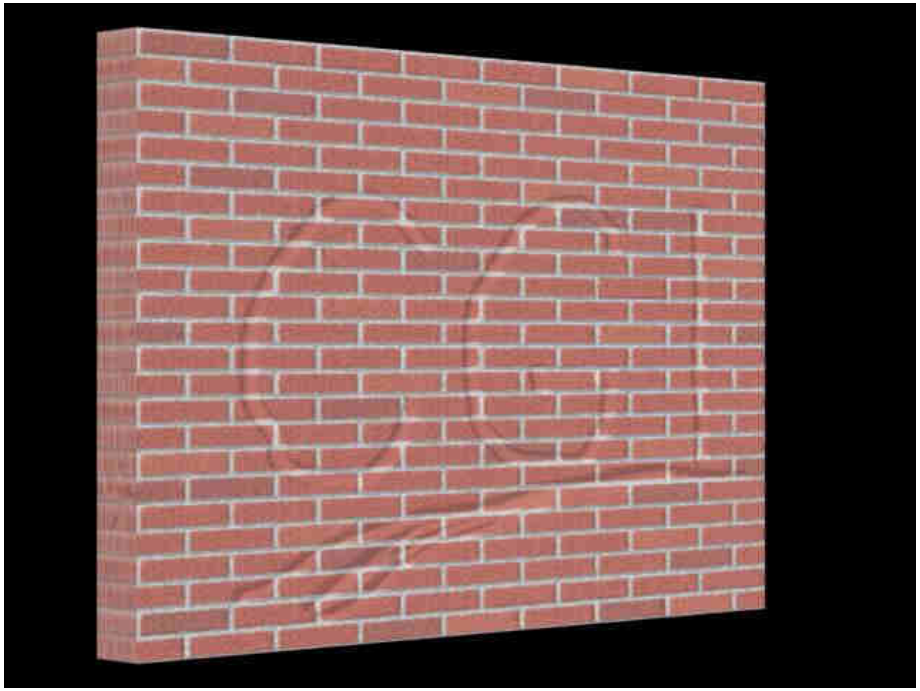
Texture Mapping (2D)



- Two steps:
  - 4 corners of the *pixel* are mapped on the surface (s, t)
  - 4 points (s, t) are mapped to texture space (u, v)
    - the resulting color is extracted from the colors of the *texels* included in the resulting area (filtering):
      - One texel color ... (bad results)
      - Weighted average of the texels colors
      - Other filtering, more powerful, exist...

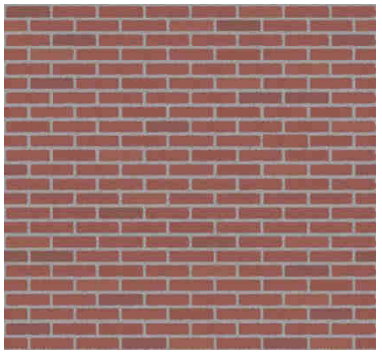
Bump Mapping Textures

- Simulation of roughness ...
  ...without increasing geometry ...

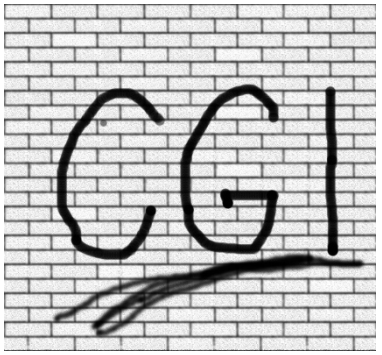


Example in 3DStudio MAX:

Mapping Texture Image



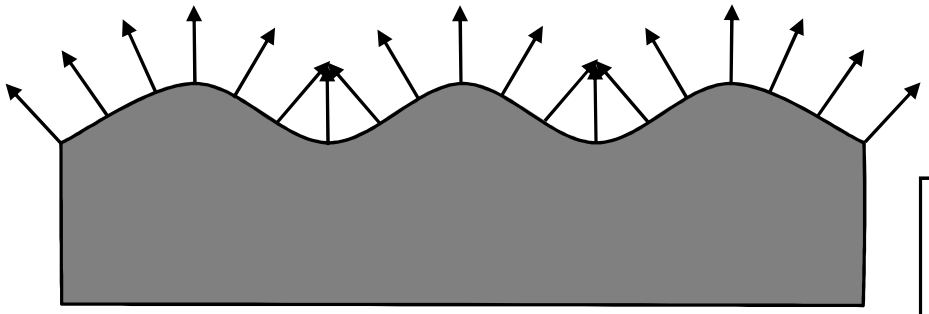
Roughness IMage



Bump Mapping Textures

Surface roughness "original"

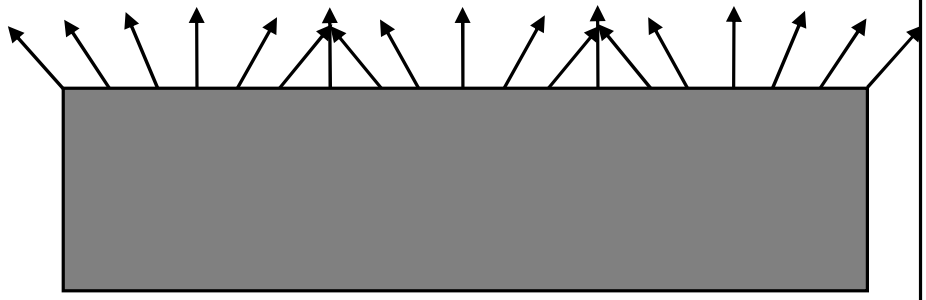
Light Source



The illumination changes, point by point, according to the inclination of their normal

Simulation of previous rough surface:

Light Source



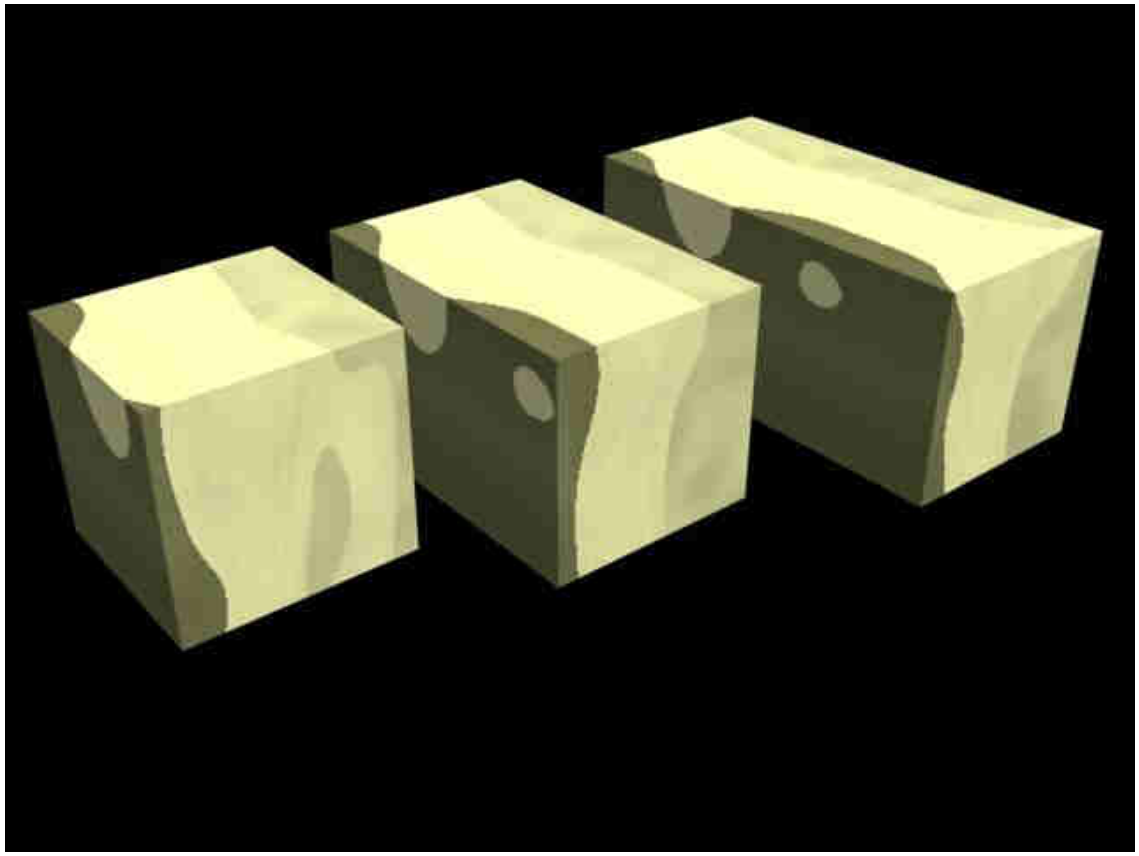
Changes in the normal direction (lighting calculation)

Illumination results similar to previous one..

BUT WITH SIMPLE GEOMETRY!

3D Textures

– Continuous evolution "inside" of the objects



- Function returns a color depending on the spatial coordinates (***x, y, z***)