Shading Interpolation

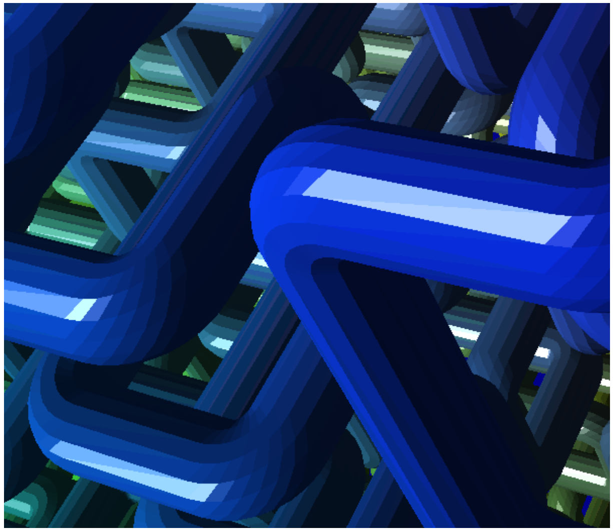
CGG MFF UK Praha

pepca@cgg.mff.cuni.cz
https://cgg.mff.cuni.cz/~pepca/

Shading Interpolation

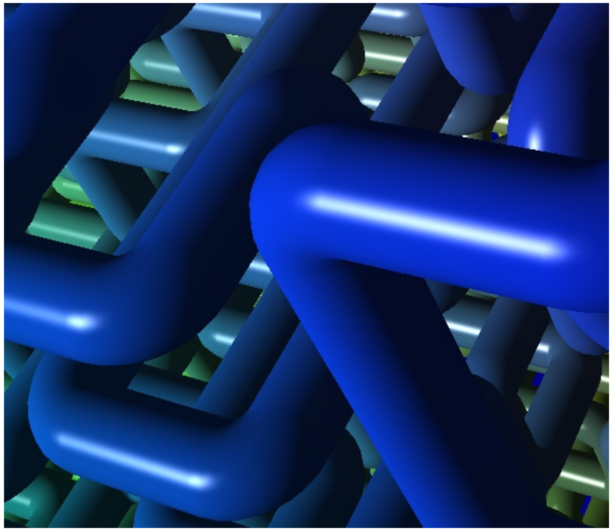
Methods to **apply shading algorithms** when displaying surfaces (B-rep)

Constant (flat) shading



Continuous shading

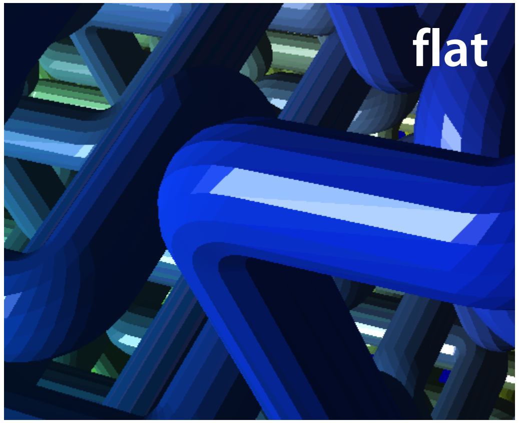
- Gouraud interpolation (colors)
- Phong interpolation (normals)



Constant Shading („Flat Shading“)

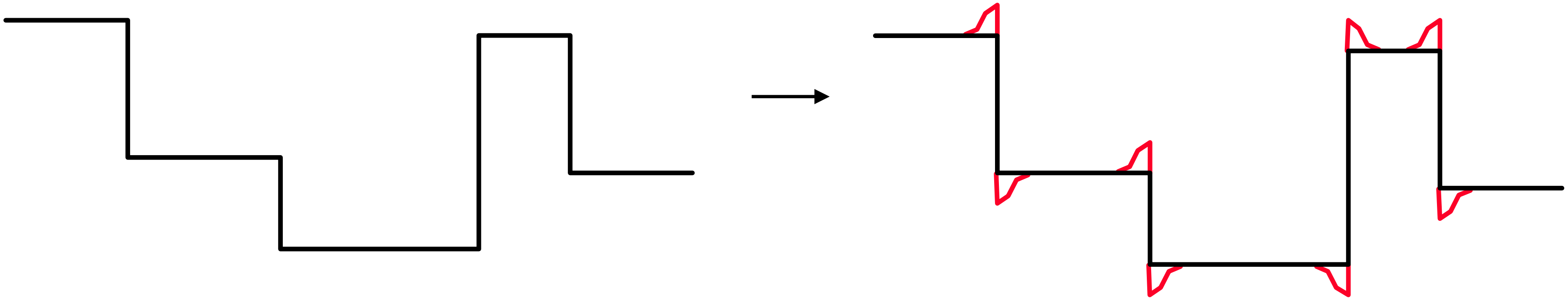
Compute **light E** once for each polygon (e.g. in the center), and fill the polygon with **a single color**

Works well for **polygonal solids**



Curved surfaces that are approximated via polygons

- artificial edges of the solid are highlighted
- this is made worse by the **Mach effect** (a human visual system feature)



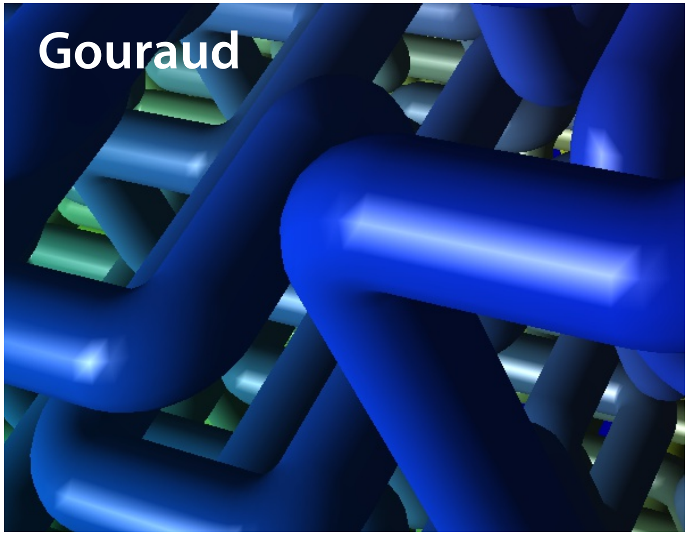
Continuous Shading

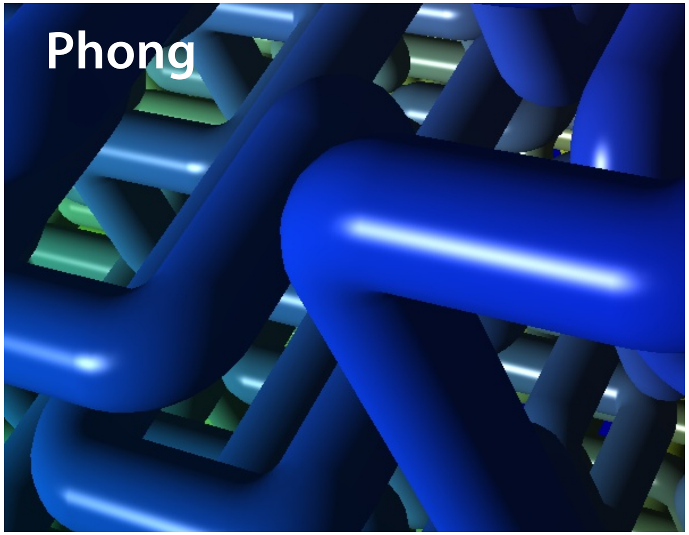
Color interpolation – **Gouraud shading**

- faster, more suited for diffuse surfaces
- HW implementations available early on (Silicon Graphics)

Normal interpolation – **Phong shading**

- slower, more accurate, suited for glossy surfaces
- nowadays avaiable in „fragment/pixel shader“ on GPU





Gouraud shading

At **polygon vertices**, we calculate the normal vector and the **color** that results from shading

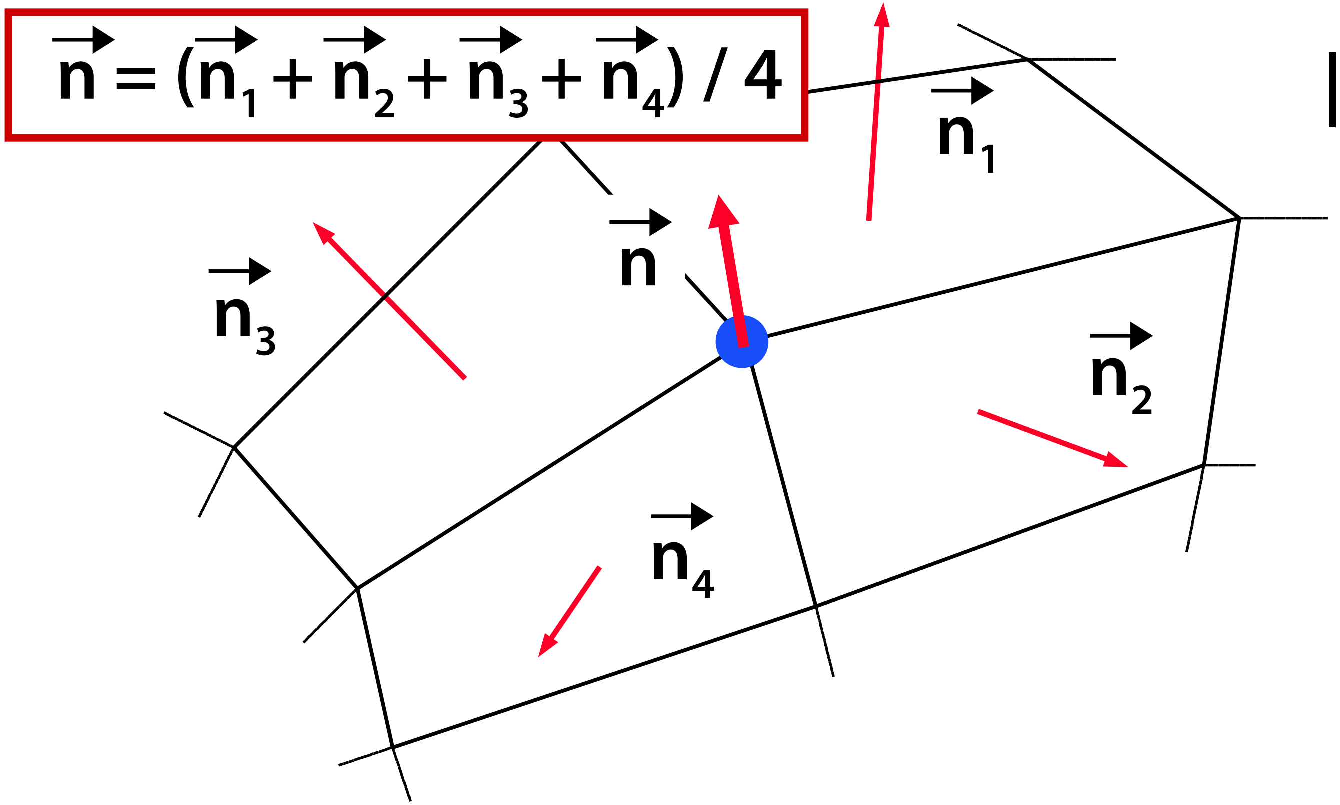
- any reflectance model can be used

Inside the polygon, we only interpolate these color values bi-linearly

- this is done during polygon filling (in GPU rasterizer)

Calculation of Vertex Normals

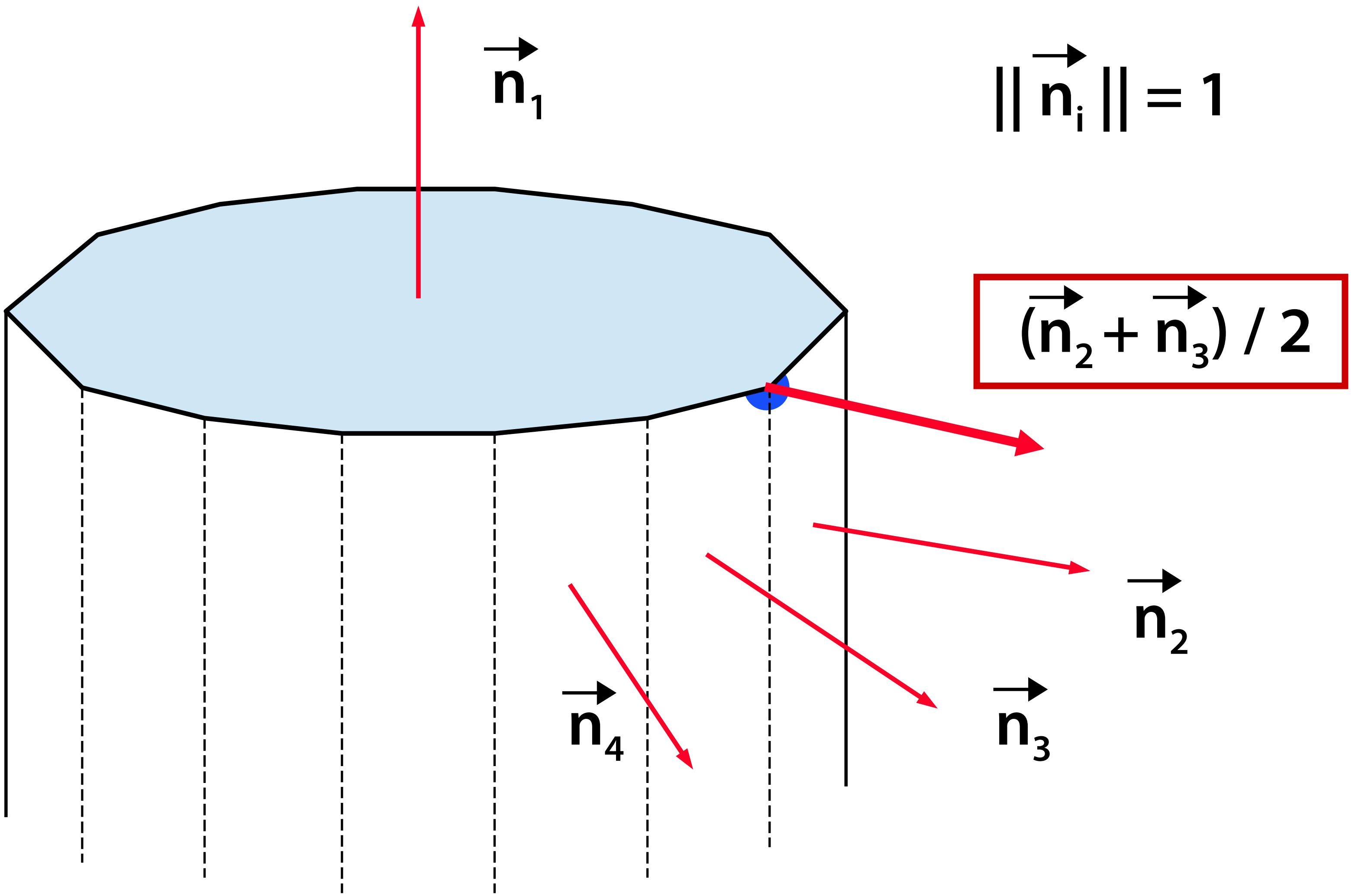
1. **Analytically** – according to neighbor polygon area
2. **Approximative** – normal average of neighbors



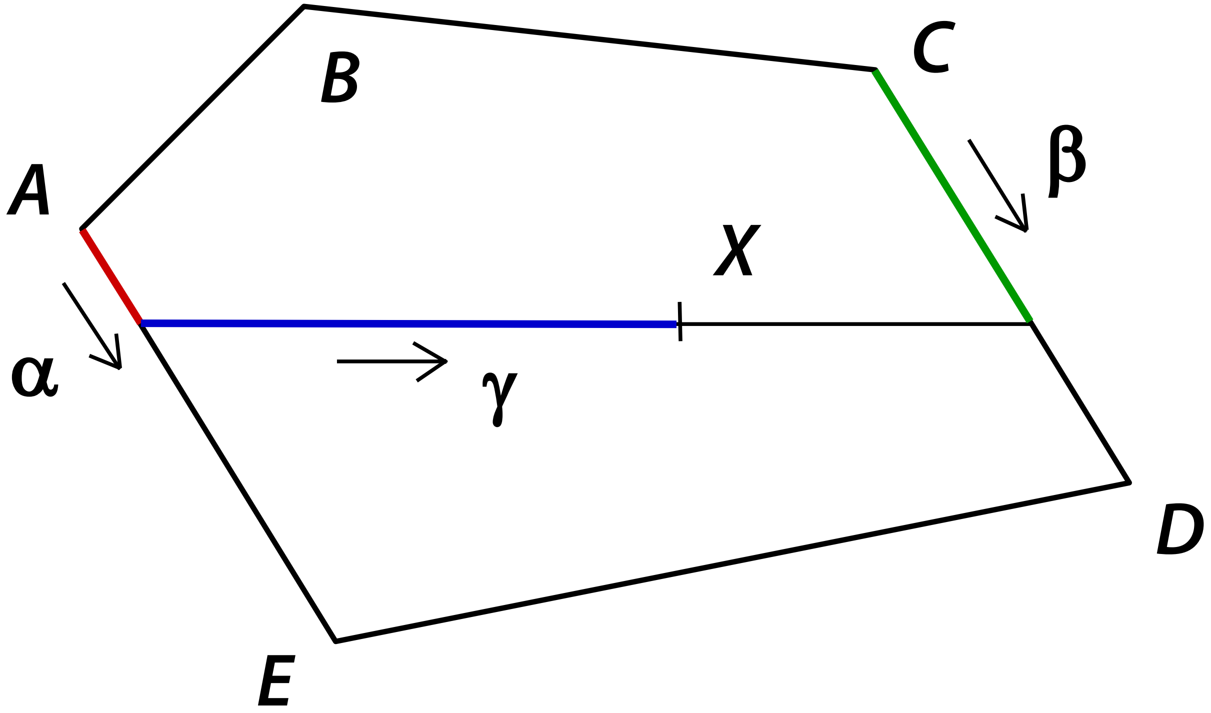
$$\vec{n} = (\vec{n}_1 + \vec{n}_2 + \vec{n}_3 + \vec{n}_4) \,/\, 4$$

$$\| \vec{n}_i \| = 1$$

Real and Internal Edges



Bi-linear interpolation



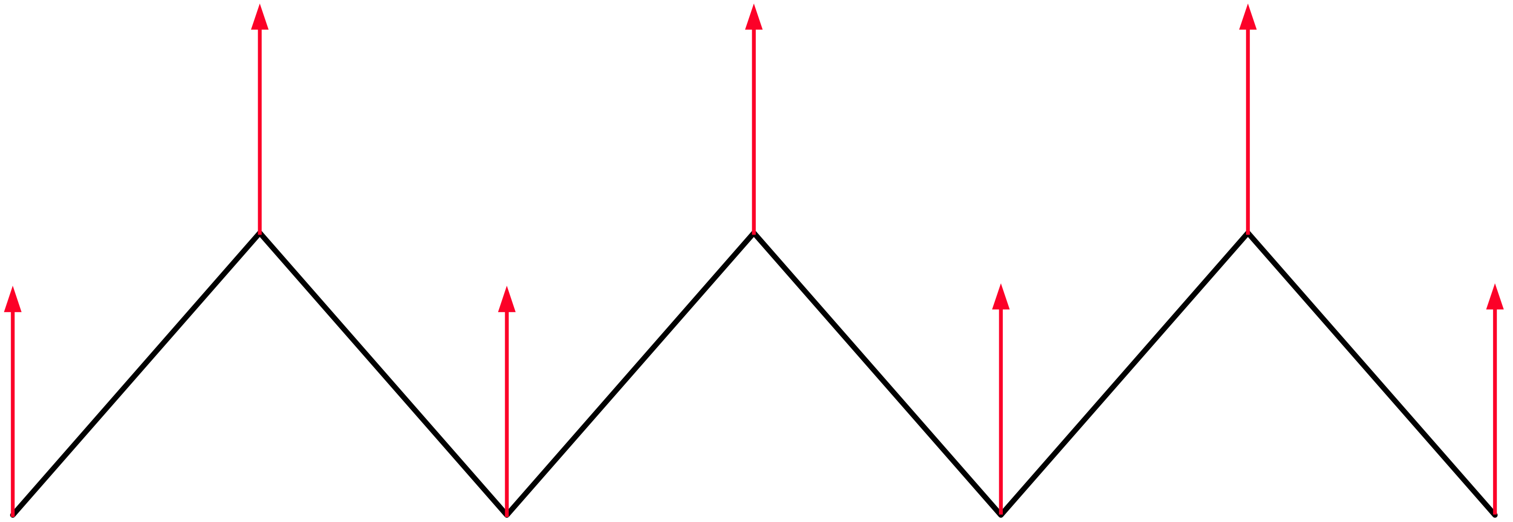
$$f_X = (1 - \gamma) \cdot [(1 - \alpha) \cdot f_A + \alpha \cdot f_E] + \gamma \cdot [(1 - \beta) \cdot f_C + \beta \cdot f_D]$$

Interpolation Issues (color)

Poorly captures **highlights** (particularly for mirror surfaces)

Not **rotationally invariant!**

Possibility of **badly calculated normals**!

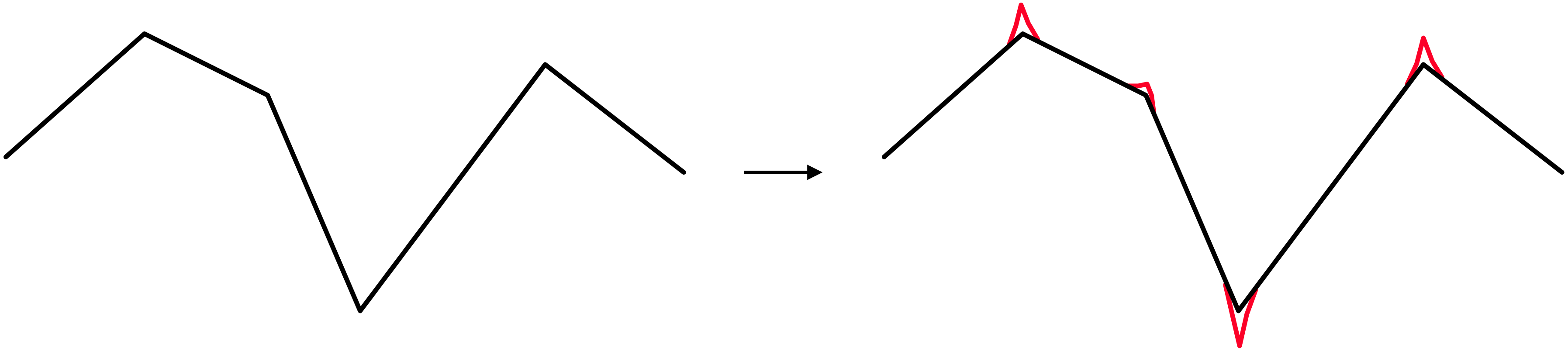


The Mach Effect (1865)

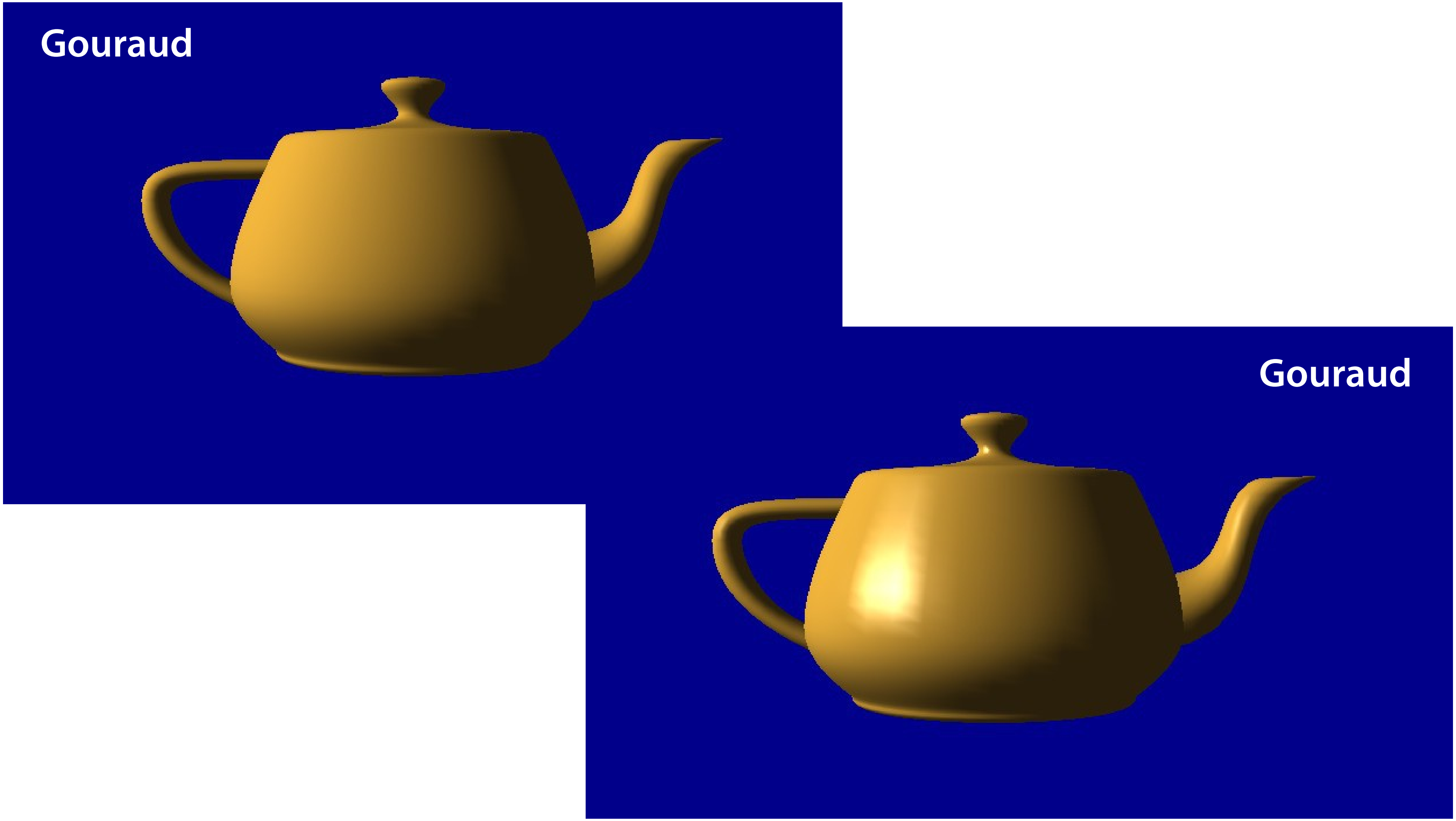
Highlight discontinuities due to **intensity** – or its **derivation**!

This is caused by **lateral inhibition** of the photoreceptors in our retina

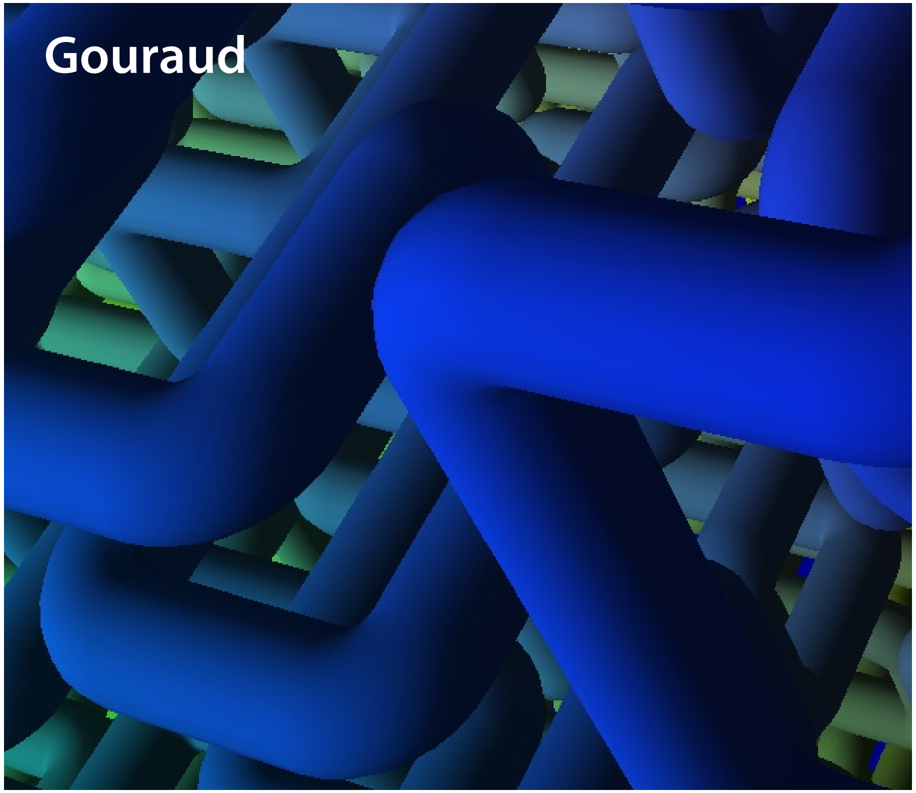
- excited cells lower the sensitivity of neighboring cells

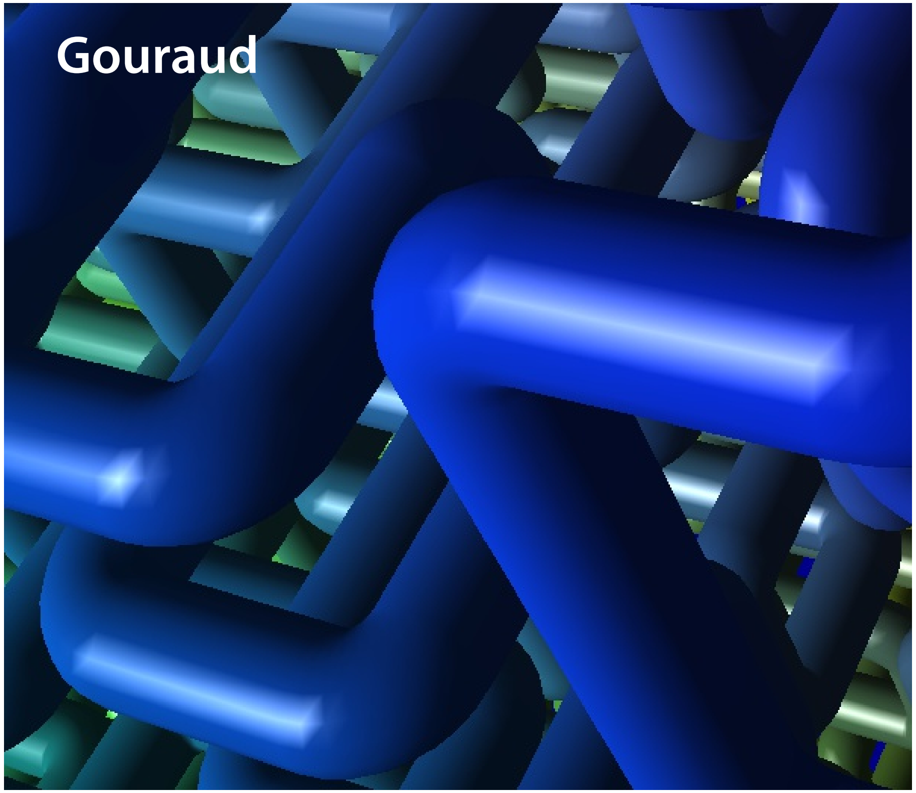


Color Interpolation – diffuse & glossy



Color Interpolation – diffuse & glossy





Phong Shading (Interpolation)

At the **polygon vertices**, we determine the correct (interpolated) **normal vectors**

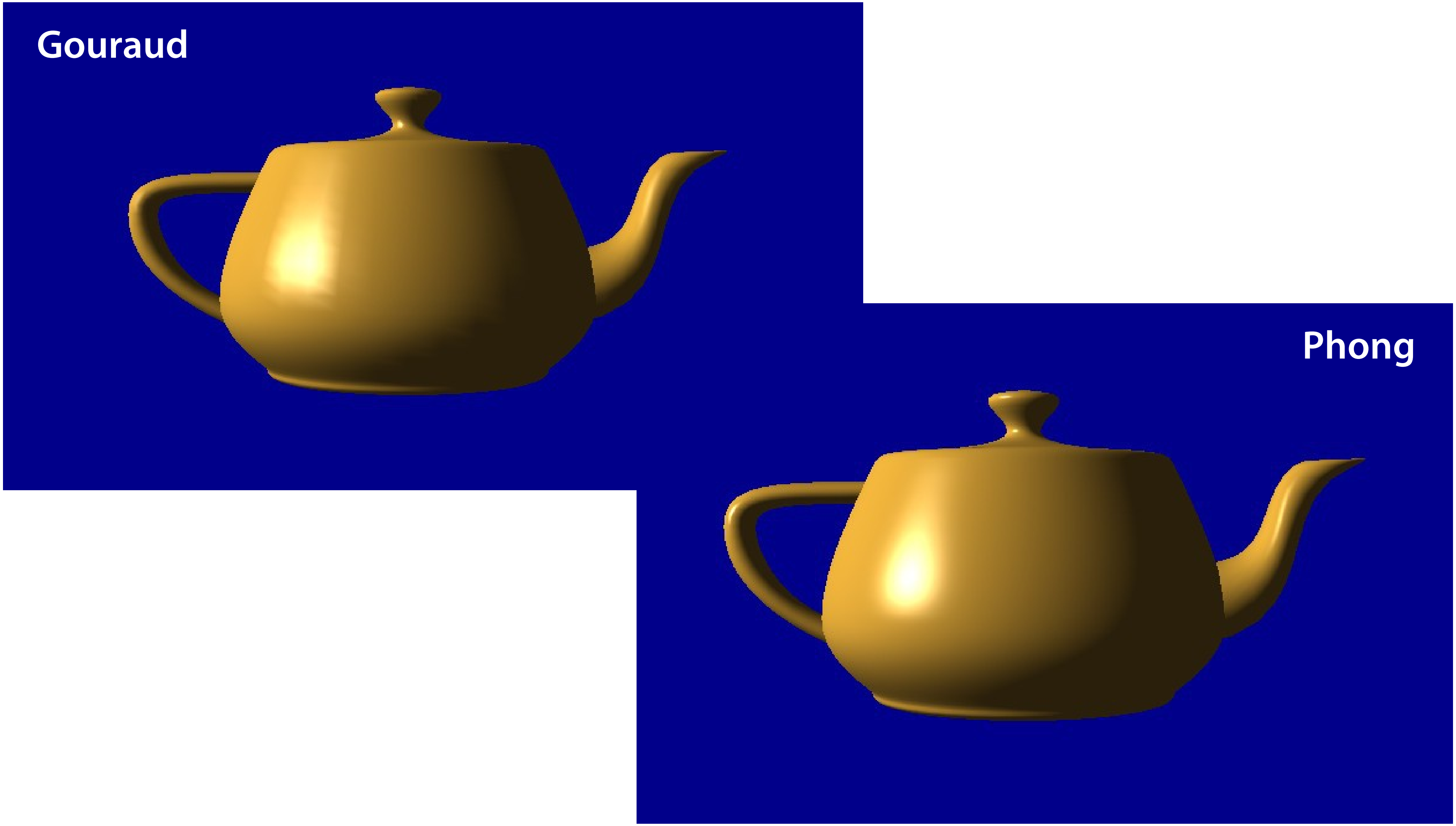
Inside the polygon, we interpolate the normal vectors via bi-linear interpolation for each pixel

- this is done in parallel with polygon filling

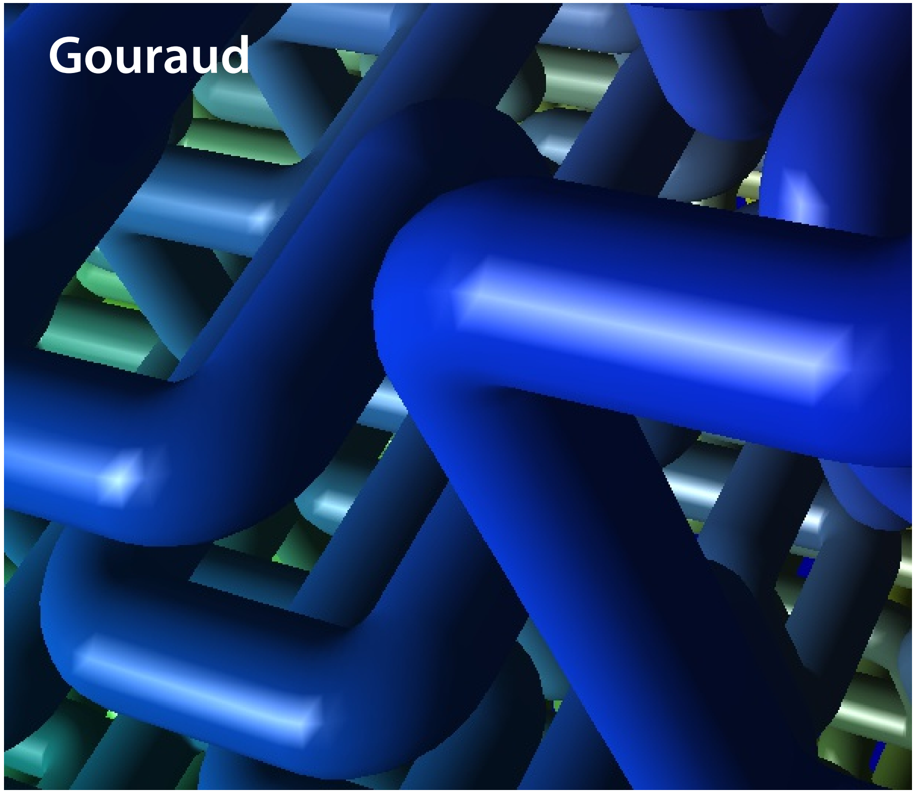
For **each internal pixel** we compute the shading (color)

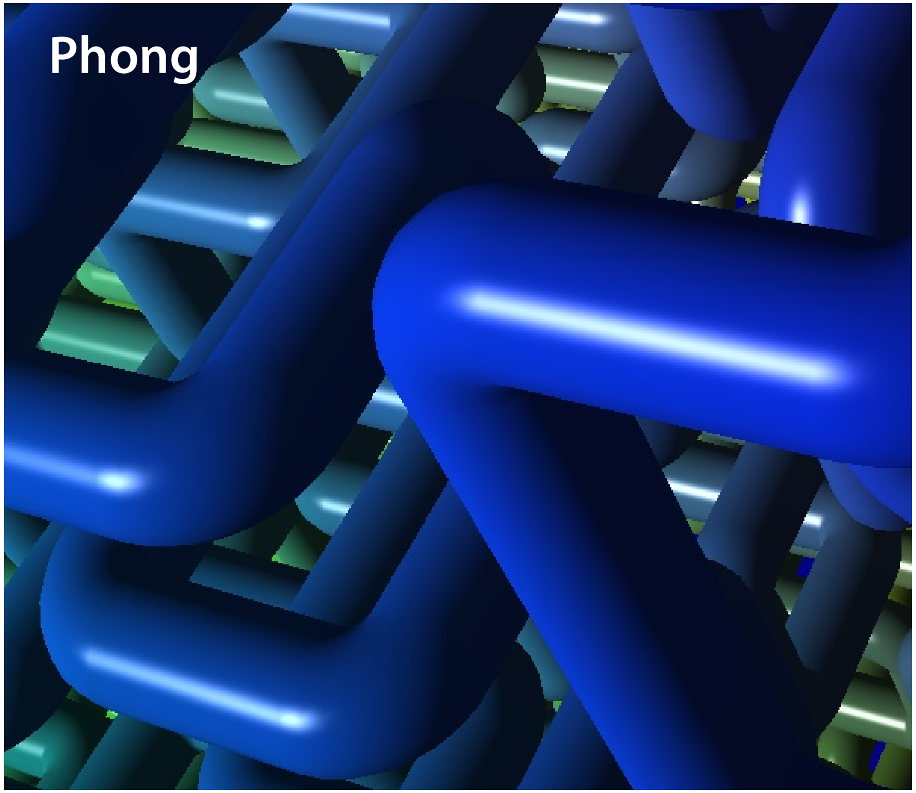
- according to the chosen shading model

Color vs. Normal Interpolation



Color vs. Normal Interpolation





Larger Computational Cost

Normals are computed for **each pixel**

- bi-linear interpolation and normalisation – this needs a **square root calculation**
- there exist approximate interpolations w/o square root

In **each pixel**, we compute the **shading model**

- dot products, square of floats, division…

Literature

J. Foley, A. van Dam, S. Feiner, J. Hughes: ***Computer Graphics, Principles and Practice***, 734-741

Jiří Žára a kol.: ***Počítačová grafika, principy a algoritmy***, 355-361