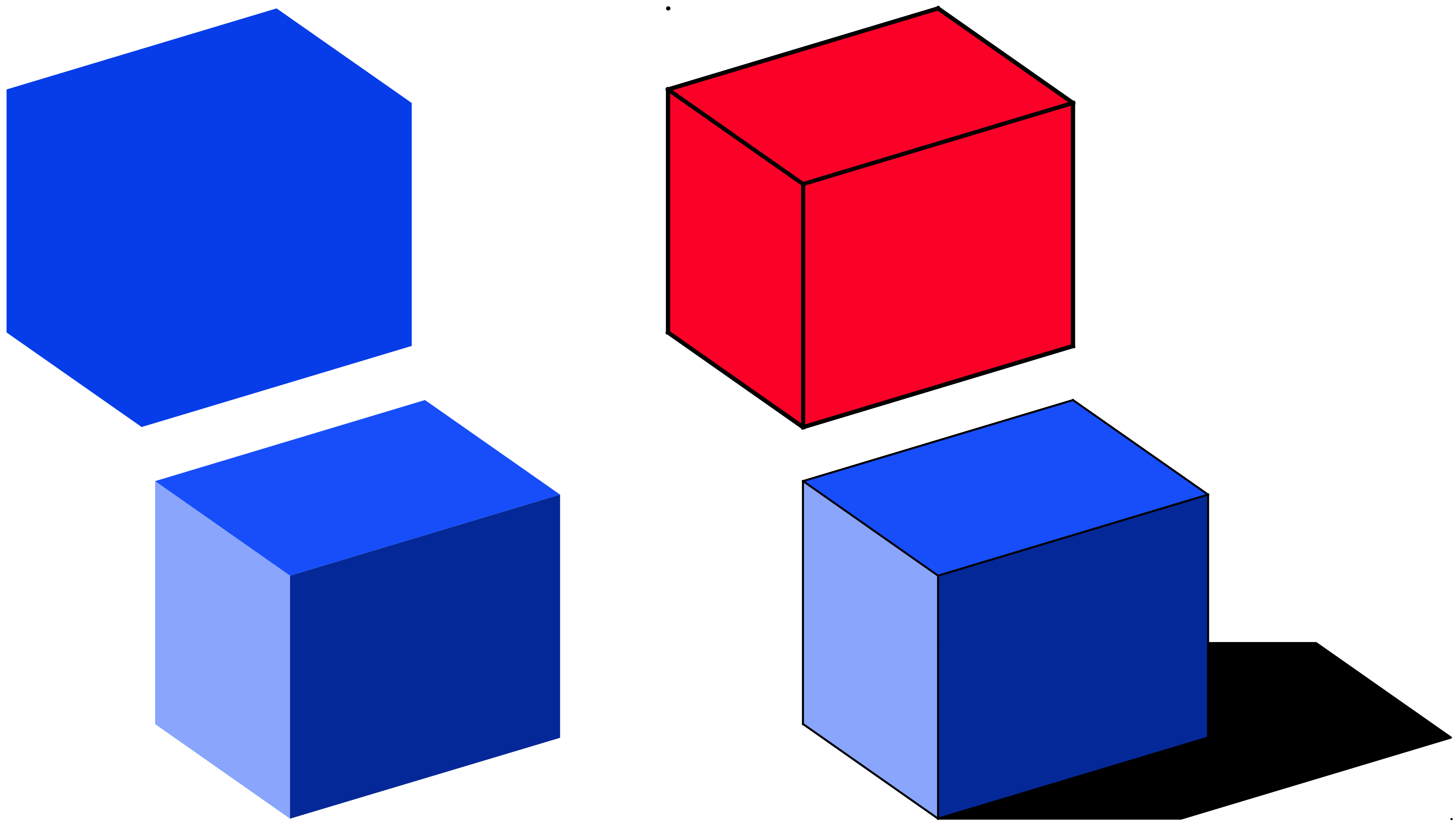
Computer Graphics Charles University

The Phong Shading Model

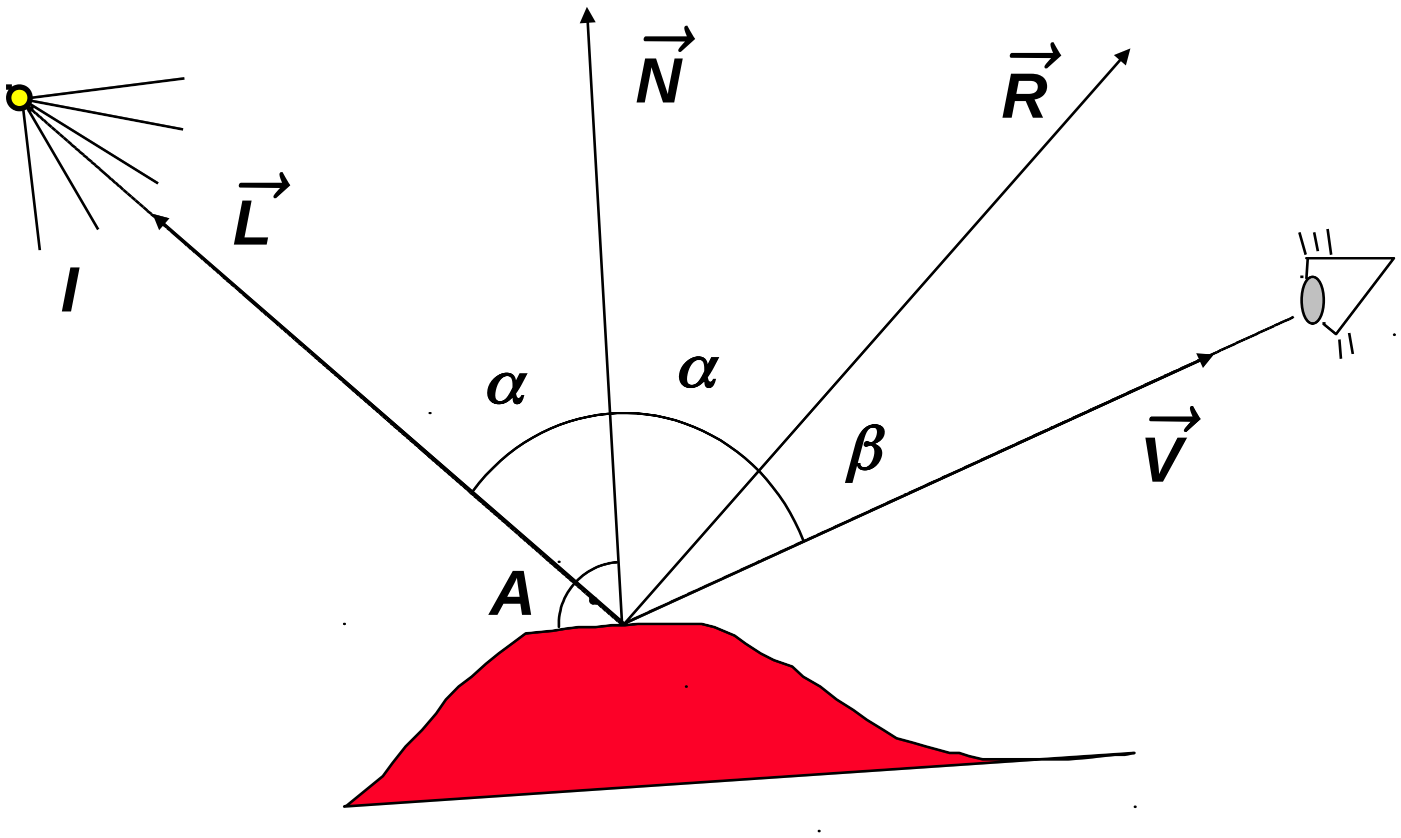
© 1995-2015 Josef Pelikán & Alexander Wilkie
CGG MFF UK Praha

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

Shading and Shadows



Shading Model



Diffuse Component E_D

- Describes an **ideally diffuse surface**

$$E_D = I_i \cdot C_D \cdot k_D \cdot \cos\alpha$$

- I_i ... light intensity
- C_D ... diffuse colour (RGB)
- k_D ... diffuse coefficient (0 .. 1)
- $\cos\alpha = L \cdot N$ ... scalar product of light direction and surface normal

Ambient Light $\mathbf{E_A}$

- **Globally constant** lighting
- Approximates / replaces **indirect light**

$$\mathbf{E_A = C_D \cdot k_A}$$

- $\mathbf{C_D}$ ... diffuse colour (RGB)
- $\mathbf{k_A}$ ... ambient coefficient (0 .. 1)

Specular Component $\mathbf{E_s}$

- Simulates a **highlight** on glossy surfaces

$$\mathbf{E_s = I_i \cdot C_s \cdot k_s \cdot \cos^h \beta}$$

- $\mathbf{C_s}$ ... highlight colour (RGB)
- $\mathbf{k_s}$ ... specular coefficient (0 .. 1)
- $\mathbf{\cos \beta = R \cdot V}$ ... dot product
- $\mathbf{h}$ ... glossiness / specularity (5 .. 500)

Integrated Model

$$\mathbf{E = E_A + E_D + E_S}$$

Colours:

- $\mathbf{C_D = C}$ ... material colour (RGB)
- $\mathbf{C_S = C_L}$ ... light colour (RGB)

Consistency:

- $\mathbf{k_A + k_D + k_S = 1}$ (to avoid overflow)

Multiple Lights

$$E = E_A + \sum_i (E_D + E_S)$$

- Calculating the **reflection vector:**

$$R = 2N\,(N \cdot L) - L$$

- Original formula for **Phong** shading for gloss:
  - Instead of constant $C_S \cdot k_S$ we can also use $W(\alpha)$ (stronger reflection for large angles)

Taking Light Distance into Account

- This should be... $\mathbf{1/d^2}$
  - Large value range (monitors cannot display it)
- Instead, use ... $\mathbf{1/(c_0 + c_1 d + c_2 d^2)}$

$$\mathbf{E = E_A + \sum_i (E_D + E_S)/(c_0 + c_1 d_i + c_2 d_i^2)}$$

Simplified Calculation

❶ **Light sources at infinity** (directional light sources)

– In the entire scene, constant light vectors $\mathbf{L_i}$

❷ **Parallel projection** (observer at infinity)

– In the entire scene, the vector $\mathbf{V}$ is constant

Simplification

❸ If both previous conditions are met, we can use $(\mathbf{R_i} \cdot \mathbf{V})^{\mathbf{h}}$ instead of $(\mathbf{H_i} \cdot \mathbf{N})^{\mathbf{2h}}$

➨ Half-way vector $\mathbf{H_i = (L_i{+}V) / |L_i{+}V|}$

– $\mathbf{H_i}$ is constant everywhere

End

Further information:

- **J. Foley, A. van Dam, S. Feiner, J. Hughes**: ***Computer Graphics, Principles and Practice***, 721-734
- **Jiří Žára a kol.**: ***Počítačová grafika***, principy a algoritmy, 343-346