Fundamentals of Computer Graphics and Image Processing

Exercise #05

Calculation of Triangle Normal

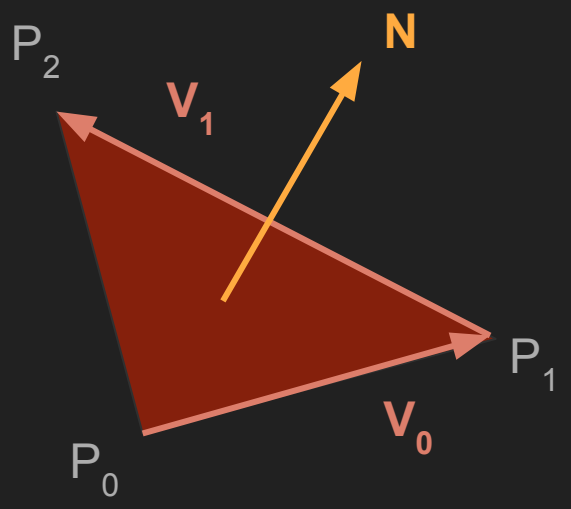
Triangle normal specifies an orientation of the triangle in 3D space

Its direction is defined by vertex order in a face record (CW or CCW)

$V_0 = P_1 - P_0$

$V_1 = P_2 - P_1$

$N = V_0 \times V_1$

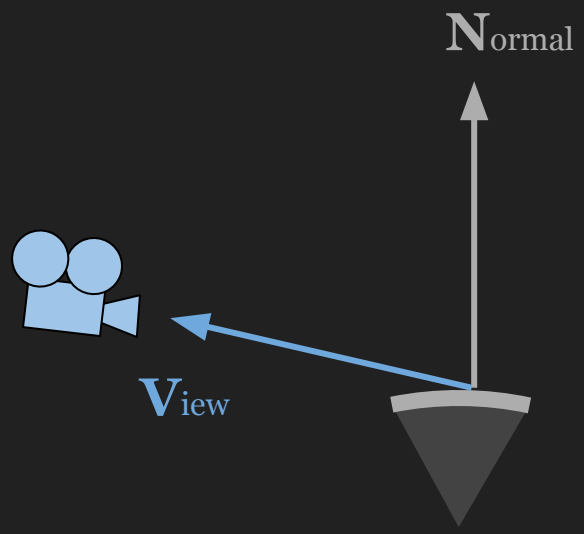


Back-Face Culling

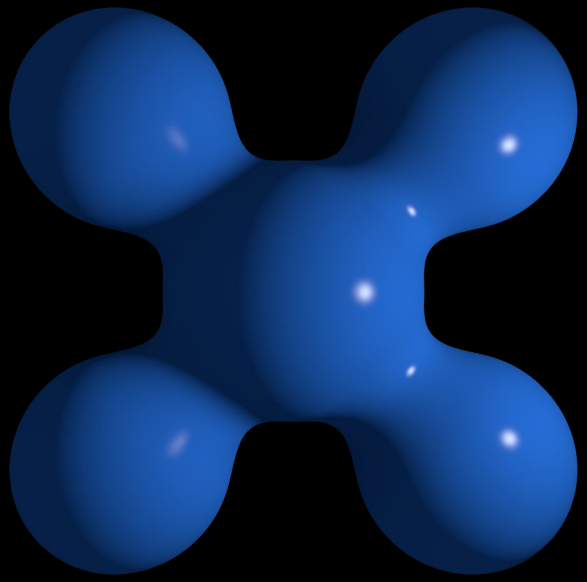
Triangle is visible to the camera if an angle of view vector and surface normal is less then 90° = their dot product is positive

IF dot(V, N) > 0 ~ visible (can be rasterized)

ELSE ~ invisible to the camera

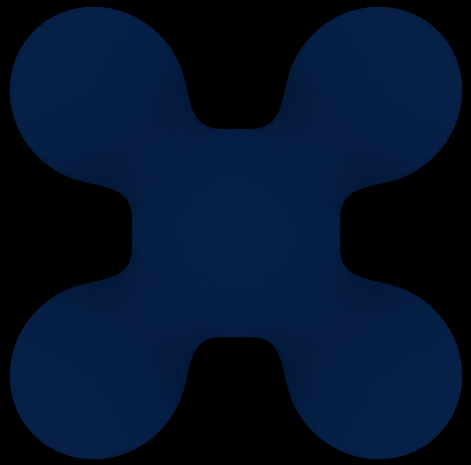


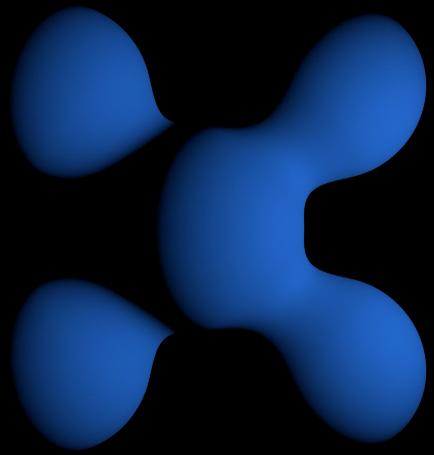
Phong Reflection Model



Final illumination of a surface can be divided into 3 components:
Ambient, Diffuse and Specular

Amount of each component in final mix depends on object's material





Phong Reflection Model

$$I = k_a I_a + \sum_L (k_d I_d + k_s I_s)$$

$$I_d = N \cdot L$$

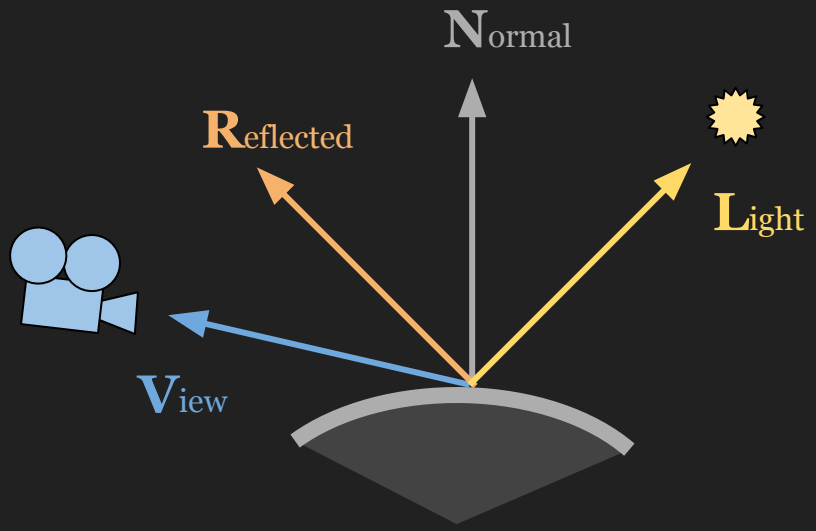
$$I_s = (R \cdot V)^h$$

Material properties

$\mathbf{k_a}, \mathbf{k_d}, \mathbf{k_s}$ - reflection constants

$\mathbf{h}$ - shininess constant

Working with normalized vectors

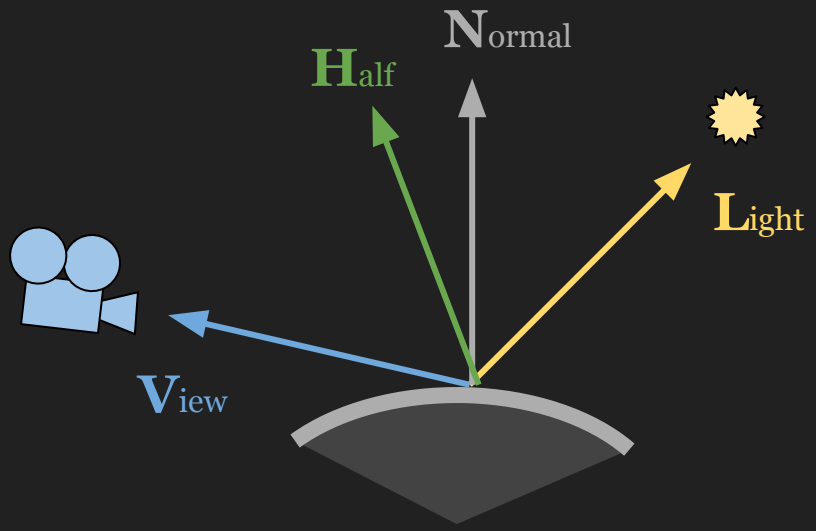


Blinn-Phong Reflection Model

$$I = k_a I_a + \sum_L (k_d I_d + k_s I_s)$$

$$I_d = N \cdot L$$

$$I_s = (H \cdot N)^h \qquad H = \frac{V + L}{2}$$



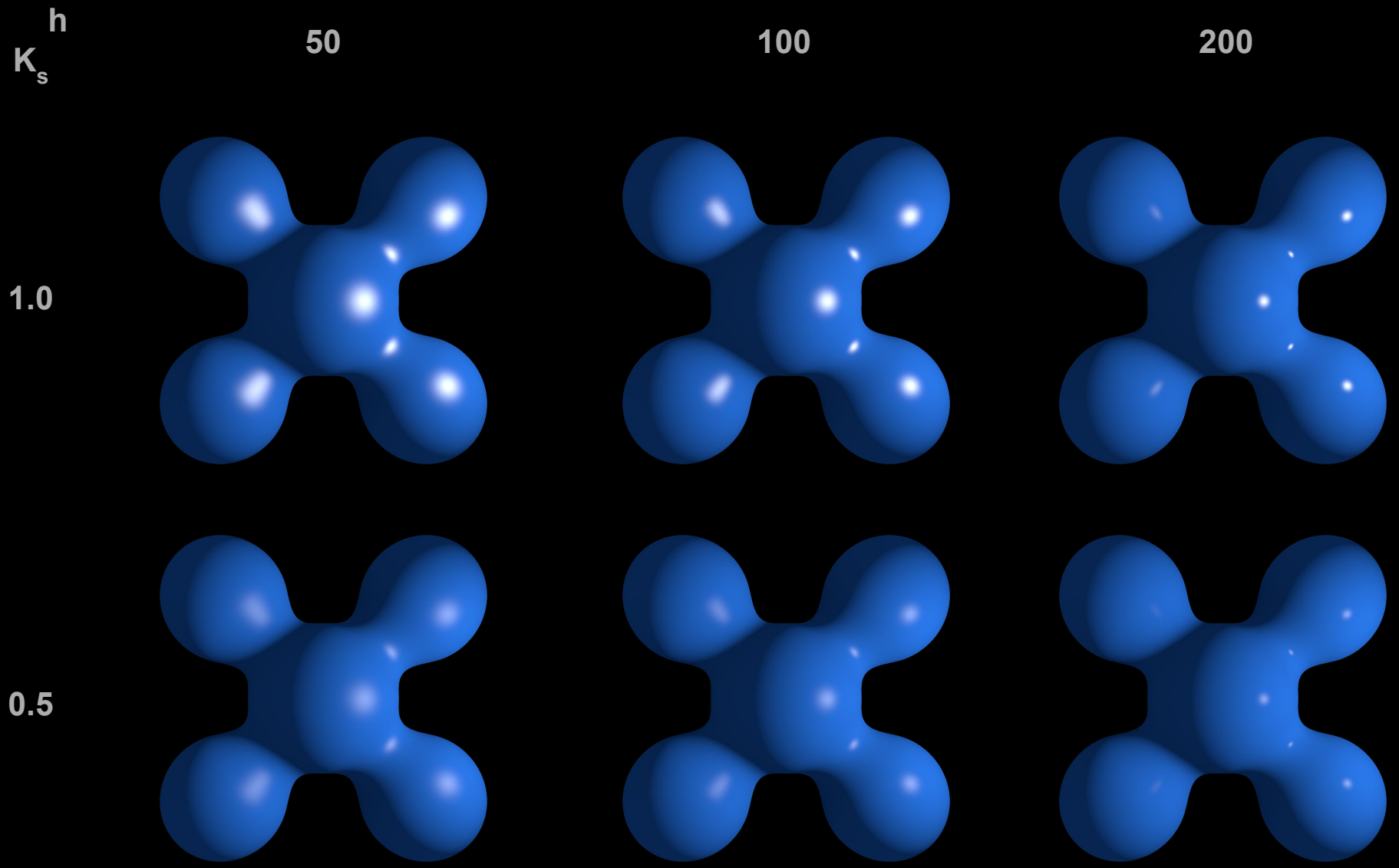
Material properties

$\mathbf{k_a}, \mathbf{k_d}, \mathbf{k_s}$ - reflection constants

$\mathbf{h}$ - shininess constant

Working with normalized vectors

Shininess Constant in Blinn-Phong



Flat Shading

Phong Shading

Phong reflection and Phong shading are not the same! Phong reflection model can be used for calculating lightning of both shading methods