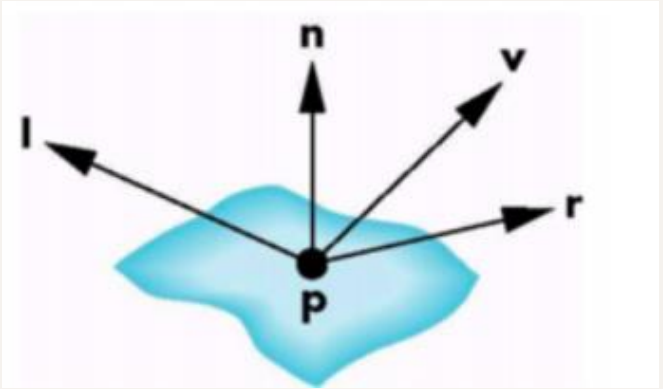
HW3

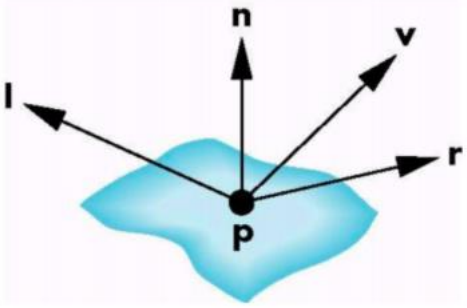
Phong Reflection Model

- L is a vector towards the light source
- N is a vector which is the normal of the point P
- V is a vector towards the camera position
- R is a vector which is the reflection of L

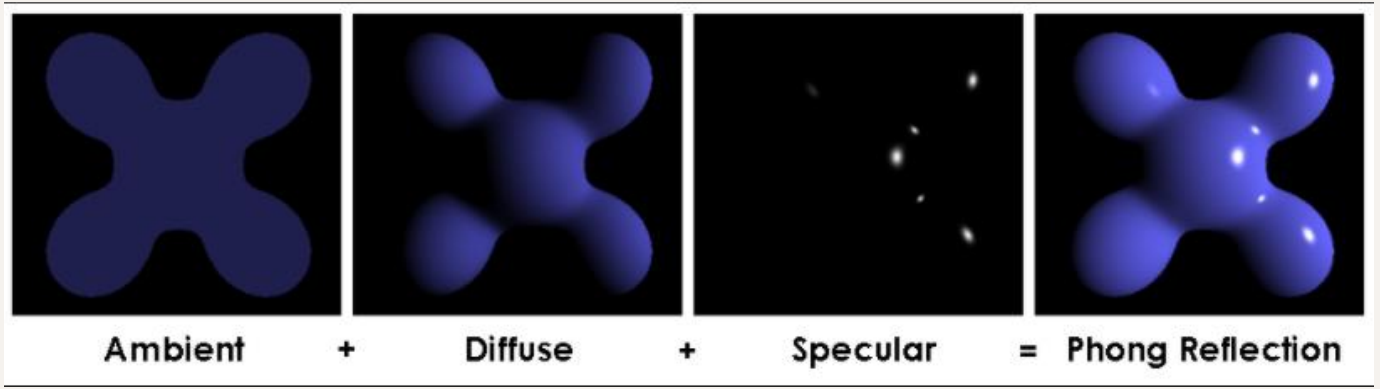


Phong Reflection Model

- $Ambient = L_a \times K_a$
- $Diffuse = L_d \times K_d \times (L \cdot N)$
- $Specular = L_s \times K_s \times (V \cdot R)^{\alpha}$



$$I = I_{ambient} + I_{diffuse} + I_{specular}$$
$$= k_a I_a + k_d I_d (l \cdot n) + k_s I_s (v \cdot r)^{\alpha}$$



Parameter

K is the reflectivity of each component of the material

- Parameters of model material:
  1. Ambient reflectivity (Ka) : 1.0 1.0 1.0
  2. Diffuse reflectivity (Kd) : 1.0 1.0 1.0
  3. Specular reflectivity (Ks) : 0.7 0.7 0.7
  4. Shininess (α) : 10.5

L is the intensity of each component of the light.

- Parameters of light:
  1. Ambient intensity (La) : 0.2 0.2 0.2
  2. Diffuse intensity (Ld) : 0.8 0.8 0.8
  3. Specular intensity (Ls) : 0.5 0.5 0.5
  4. Position : (10, 10, 10)

Blinn-Phong shading

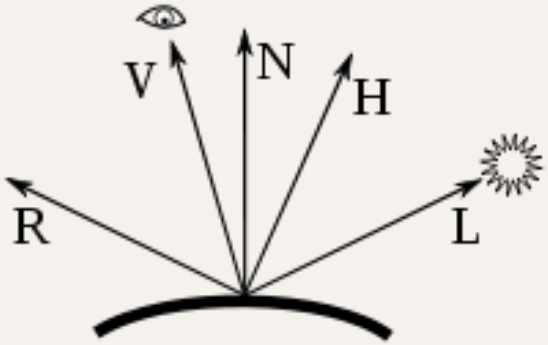
- Calculate a halfway vector between the viewer and light vectors

$$H = \frac{L + V}{\|L + V\|}$$

- Use $N \cdot H$ to replace $\mathrm{R} \cdot V$, $Specular = L_s \times K_s \times (\mathrm{N} \cdot \mathrm{H})^{\alpha}$

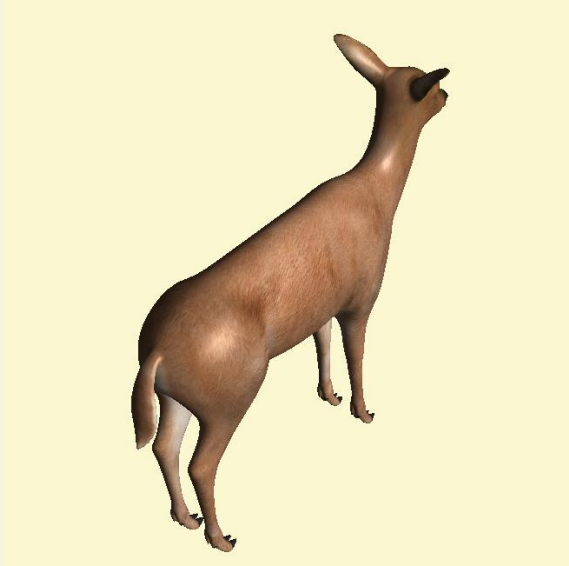
Phong shading

Blinn-Phong shading



Gouraud shading

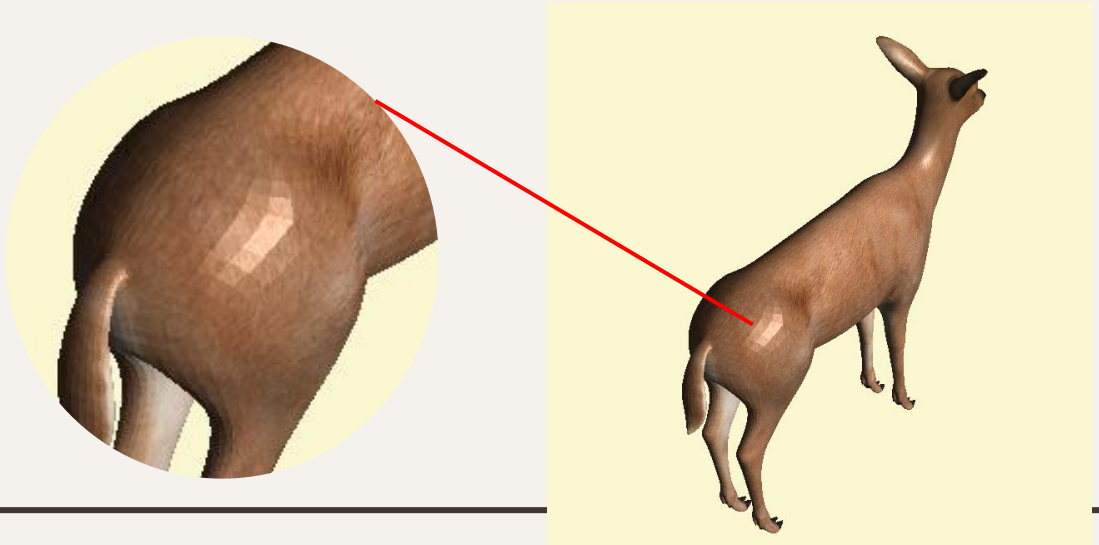
- Implement the Phong lighting model at each vertex
- Define normals at each vertex and use them to calculate lighting



Flat shading

- Define a normal for each polygon. Then, calculate lighting for each polygon.
- The color is the same for all points of each polygon.
- MUST use a geometry shader to compute the normals for each of the primitives.

  CANNOT use keyword “flat” to skip interpolation.

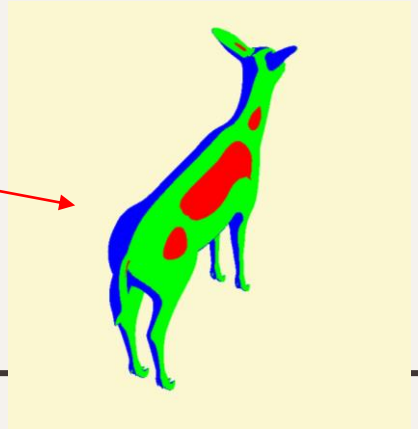


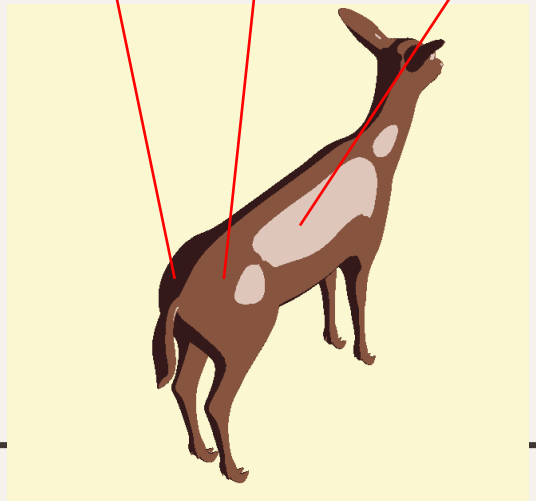
Toon shading

- Calculate the angle between the light and normal vector
- If the angle is greater than 90°, give low intensity or a dark color
- If it has strong specular, give high intensity or a light color
- Else, give medium intensity
- You can decide the thresholds and colors yourself,

but must define at least 3 levels and the result is reasonable

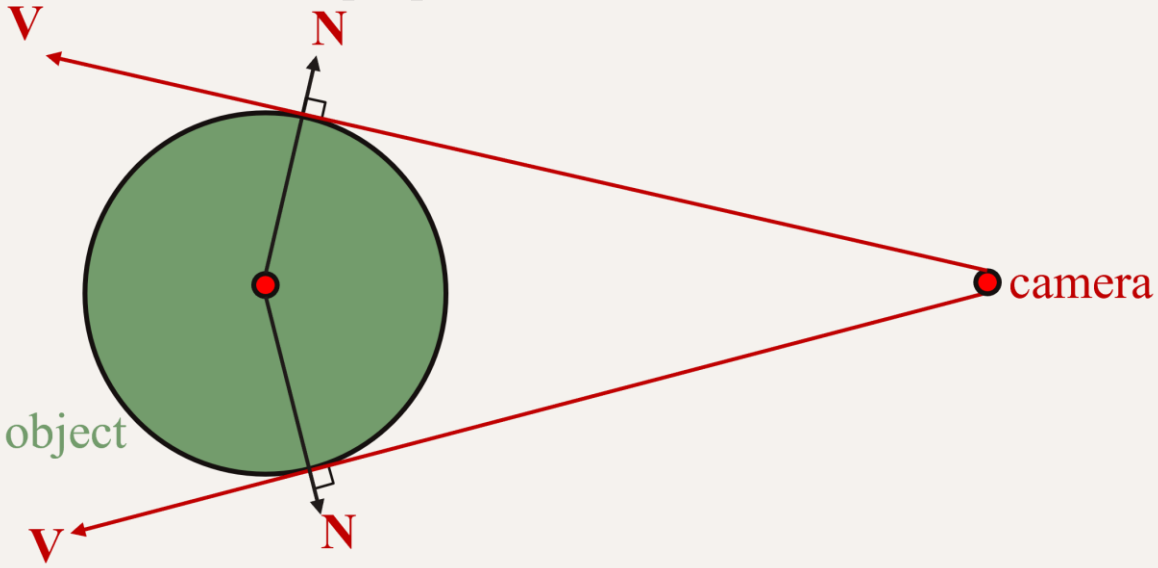
This is not allowed

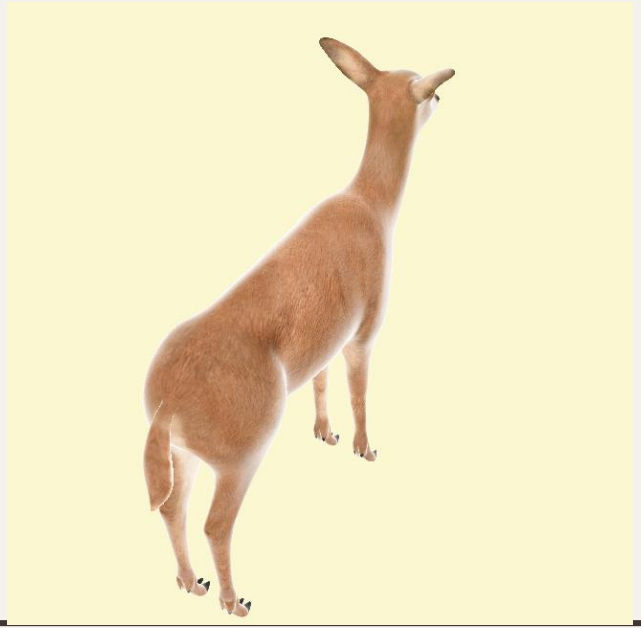




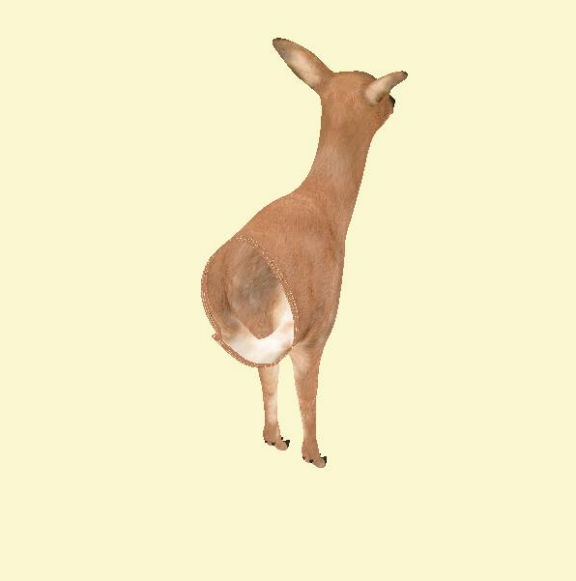
Advanced : Border effect

- How to determine if a point is on the border of an object ?
  - Calculate the angle between the normal(N) and view(V) vectors.
  - If V is perpendicular to N, then it is a border





Advanced : Dissolve effect



Homework3

- Basic :

Press "1" : Blinn-Phong shading

Press “2" : Gouraud shading

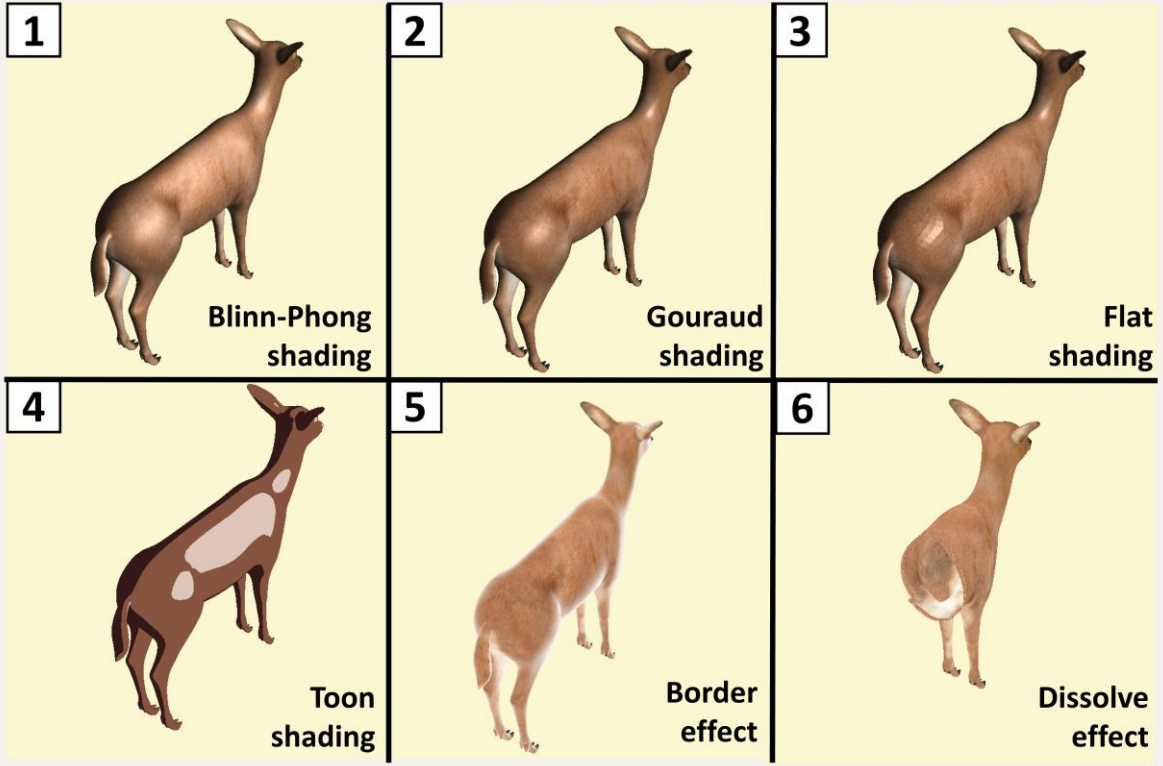
Press “3" : Flat shading

Press “4" : Toon shading

- Advanced :

Press “5" : Border effects

Press “6" : Dissolve effects



Homework3 - score

1. Create shaders and programs you need and can switch them correctly (5%)
2. Create all variable and pass them to shaders through Uniform (5%)
3. Implement Blinn-Phong shading via shader (15%)
4. Implement Gouraud shading via shader (15%)
5. Implement Flat shading via shader (15%)

   # MUST use the geometry shader to calculate the normal
6. Implement Toon shading via shader (15%)

   # At least define 3 levels and the result is reasonable
7. Report (20%)
8. Advanced - Implement Border effect shading via shader (5%)
9. Advanced - Implement Dissolve effect shading via shader (5%)

Homework3 - Report

- Please specify your name and student ID in the report.
- Explain how you implement the above shading/effects.

  (ex: how I get the vector L. I do dot(L, N) for what……etc.)
- Describe the problems you met and how you solved them.
- File name: hw3_report_studentID.pdf

Homework3 - Submission

- Deadline: 2023/12/18 23:59:59
- 10% penalty for each week late
- Pack your report(hw3_report_studentID.pdf) and project in a zip file.

  File name: hw3_studentID.zip
- If your uploading format doesn’t match our requirement, there will be penalty to your score. (-5%)

Homework3 - Files to be implemented

- main.cpp
- shaders/Blinn-Phong.vert
- shaders/ Blinn-Phong.frag
- shaders/Gouraud.vert
- shaders/Gouraud.frag
- shaders/Flat.vert
- shaders/Flat.geom
- shaders/Flat.frag
- shaders/Toon.vert
- shaders/Toon.frag

Advanced :

- shaders/Border.vert
- shaders/ Border.frag
- shaders/Dissolve.vert
- shaders/Dissolve.frag