

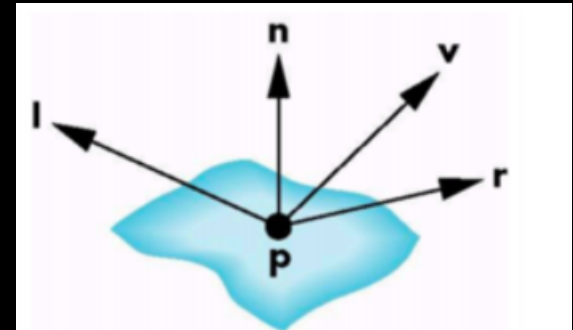
HW3

2020 Introduction to Computer Graphics

How to determine light intensity

- We can simply use the included angle of the reflection and view vectors.

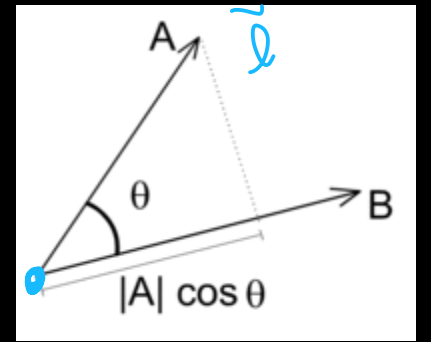
- L is a vector towards the light source
- V is a vector towards the camera position
- R is a vector which is the reflection of L
- N is a vector which is the normal of the point P



如何在畫圖時簡單
決定光強度
⇒ 用夾角

How to determine light intensity

光源
打到的点



法

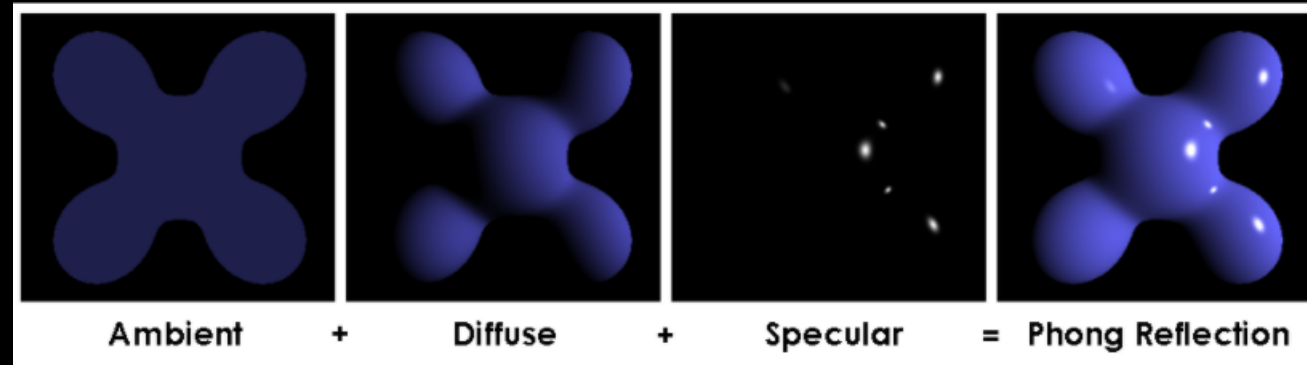
- If two vectors are unit vectors. Then we can get $\cos \theta$ by doing dot products of the two vectors.

$$A \cdot B = |A| |B| \cos \theta$$

- The smaller θ is, the larger $\cos \theta$ is. According to the Phong reflection model, we can determine the light intensity based on $\cos \theta$.
- If $\cos \theta < 0$, θ must bigger than 90° . In this case, this position cannot be illuminated.

Phong shading

物体
表面反射
系数



K is the reflectivity of each component of the material

- Parameters of model material:

1. Ambient reflectivity (K_a) : 1 1 1
2. Diffuse reflectivity (K_d) : 1 1 1
3. Specular reflectivity (K_s) : 1 1 1

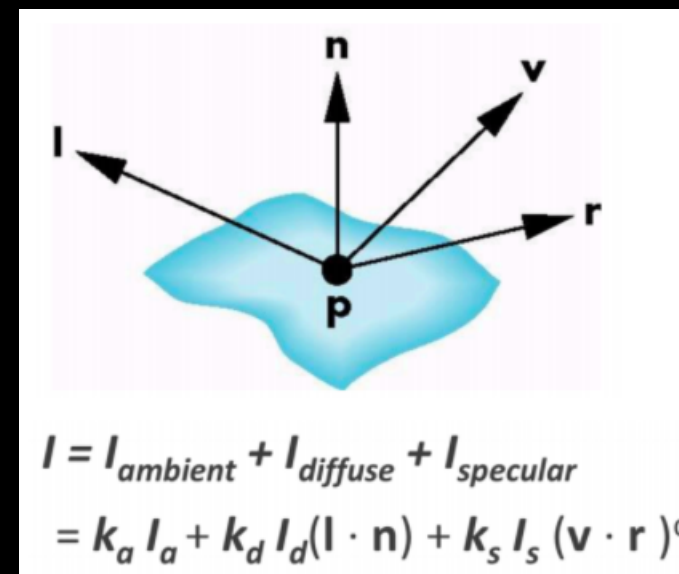
L is the intensity of each component of the light.

- Parameters of light:

light 系数

1. Ambient intensity (L_a) : 0.2 0.2 0.2
2. Diffuse intensity (L_d) : 0.8 0.8 0.8
3. Specular intensity (L_s) : 0.5 0.5 0.5
4. gloss (Specular shininess factor) : 25

α is the glossiness of the material. α 越大 \Rightarrow 表面材质反射 \uparrow



Phong shading - pseudocode

```
void main()
{
    object_color = texture2D(Texture, texcoord);

    ambient = La * Ka * object_color;
    diffuse = Ld * Kd * object_color * dot(L,N); // must > 0
    specular = Ls * Ks * pow(dot(V,R), gloss);

    color = ambient + diffuse + specular;
}
```

Handwritten note: dot(L,N) must be > 0 (dot product)

Toon shading - pseudocode

```
void main()
```

```
{
```

```
    object_color = texture2D(Texture, texcoord);
```

```
    Decide a level by calculating the included angles between the Light and normal vectors
```

```
    if (level > 0.95) intensity = 1;  
    else if (level > 0.75) intensity = 0.8;  
    else if (level > 0.50) intensity = 0.6;  
    else if (level > 0.25) intensity = 0.4;  
    else intensity = 0.2;
```

1
p
5
大色塊

→ 直接把顏色分開，
這樣看起來有一塊一塊
大色塊

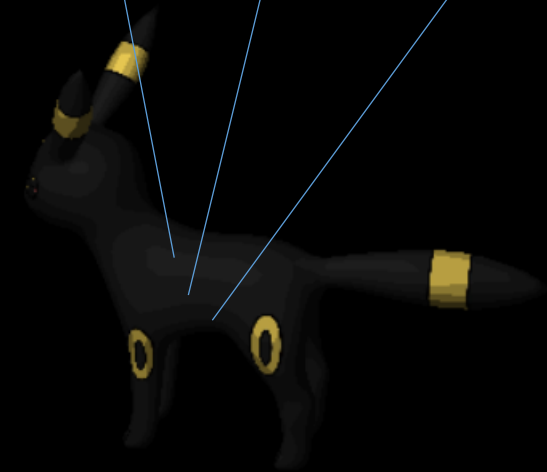
```
    Color = Kd * object_color * intensity ;
```

```
}
```

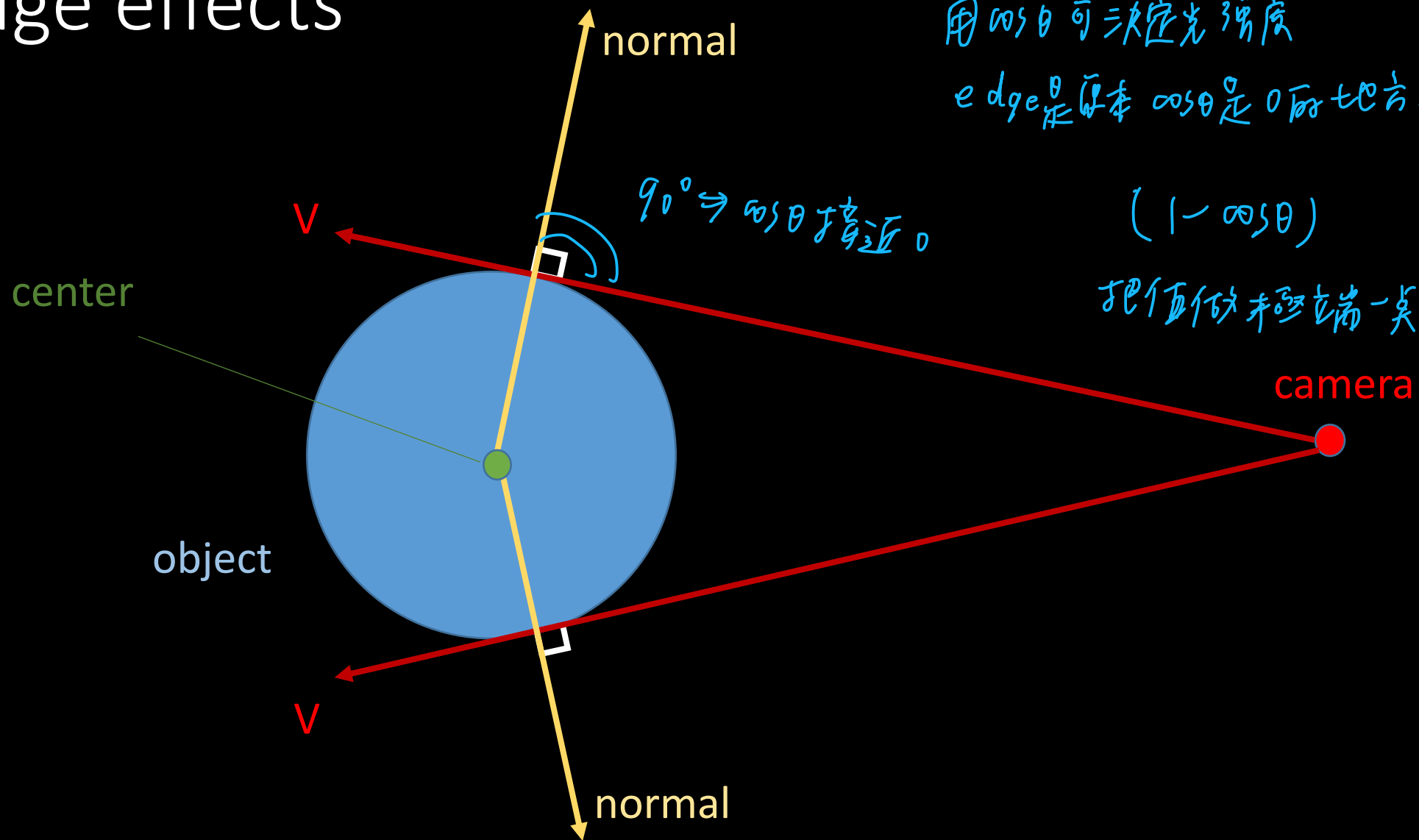
Level > 0.95

0.95 > Level > 0.75

0.75 > Level > 0.5



Edge effects



reminder

顏色在不同 space 上會不同

map

是讓 world space 和 model space 到同一個空間

要做正確的 space 轉換

the light position, camera position and the model position might not be in the same space. If so, you need to do some space conversion. Therefore you might need some inverse matrices to do that.

For example, to retrieve inverse model matrix, you can do inverse transformation and get current matrix.

```
normal = normalize((M * vec4(in_normal, 1.0)).xyz);
```

→ 不要做!

This will work in this homework, but actually it is the **wrong** way to do the space conversion.

↓ 'normal 可能會歪掉

Homework 3 - Shading



Homework 3

- Goal :
 1. Phong shading
 2. Toon shading
 3. Edge effects



Phong shading



Toon shading



Edge effects

Homework 3 (配分)

1. create shaders and programs you need and can switch them correctly.(5%)
2. Pass all variable to shaders and trigger by Uniform(5%)
3. Implement **Phong shading** via shader (40%)
4. Implement **Toon shading** via shader(30%)
 - # at least define 5 levels.
5. Implement **Edge effects** via shader(10%)
 - # must clearly see the edge
 - # The color of the edge is not specified
6. Report (10%)

This is not allowed



Homework 3 (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 $\text{dot}(L, N)$ for what.....etc.)
↳ 向量的向量... 如何決定, 為什麼要做內積...
- Describe the problems you met and how you solved them.

Homework 3 (繳交規則)

1. DeadLine: 2020/ 12 / 22 23: 59:59
2. Penalty of 10% of the value of the assignment per late week.

If you submit your homework late, the score will be discounted.

submit between (12/23 - 12/29) : Your final score * 0.9

submit between (12/30 - 1/5) : Your final score * 0.8

submit after 1/6 : Your final score * 0.7

Restrictions !!

- Your GLSL version should `>= #version 330`
- Deprecated shader syntaxes are not allowed, e.g. attribute, varying
- You are only allowed to pass uniform data to shader using glUniform* series function
- Using built-in uniform variables in shader is forbidden!
 - (That is, you **cannot** use gl_ModelViewMatrix or gl_NormalMatrix ...etc)
 - The only gl_XXX term should be in your shader code is gl_Position.
- **All the variable used in shaders must use uniform to pass , cannot directly define in shader.**

Upload Format

1. If your uploading format doesn't match our requirement, there will be penalty to your score. (-5%)
2. Please hand in the whole **project file** and **report** (.pdf) as STUDENTID_Name.zip to e3 platform.
e.g. 0716XXX_王小明.zip

#project file要載下來就可以demo