

CPT205W09_光照与材质

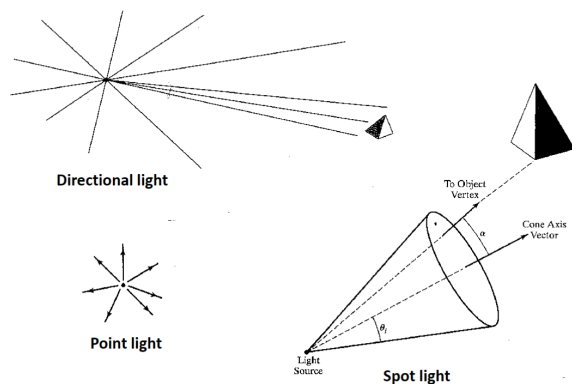
背景

通过生成对象的透视投影并将自然照明效果应用在可见表面可获得场景的逼真显示。

要实现照片级的真实感还涉及表面材料属性的表示和良好的物理光照效果（反射，透明度，表面纹理，阴影）

光源

可设置许多属性，如位置，方向，颜色，形状，反射率。为每个RGB分量设置一个值，以描述颜色的强度。



点光源 (Point light)

最简单的物体模型，它以三个 RGB 分量指定的单一颜色发射辐射能量。

如果大型光源不太靠近场景，则可以将其视为点光源。光线是沿着从光源位置径向发散的路径产生的。

定向光 (Directional light)

无限远/定向光（光束是平行的）

距离场景非常远的大型光源（例如太阳）也可以近似为点光源，但其方向效果几乎没有变化。

聚光灯 (Spot lights)

通过修改局部光源以产生聚光灯光束。

如果物体超出光源的方向限制，则该光源会将其排除在照明之外。通过分配矢量方向和从矢量方向测量的角度限制以及位置和颜色来设置聚光光源。

表面光照效果 (Surface lighting effects)

尽管光源提供单一的频率分布，但环境、漫反射和镜面反射分量可能不同。例如，如果我们在一个红墙的房间里有一盏白光，那么散射光往往是红色的，尽管直接照射物体的光是白色的。

■ 环境光 (Ambient light)

效果为一般背景非定向照明。每个对象都使用其固有的强度进行显示，即非反射、自发光对象的世界。

当环境光照射到表面时，它会向各个方向均匀散射。

房间中的背光具有较大的环境成分，因为到达眼睛的大部分光线首先从许多表面反射回来。户外聚光灯的环境成分很小；大部分光线沿同一方向传播，并且由于位于室外，因此从其他物体反射后到达眼睛的光线非常少。

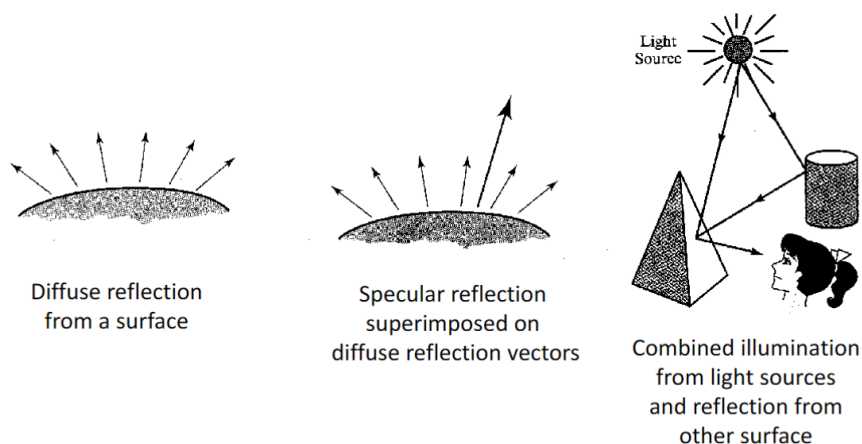
■ 漫反射 (Diffuse reflectance)

一旦它撞到表面，它就会均匀地分散到各个方向，因此无论眼睛位于何处，它看起来都一样亮。来自特定位置或方向的任何光线都可能具有漫反射组件。粗糙或颗粒状表面往往会向各个方向散射反射光（称为漫反射）。影响照度的因素是点光源、源强度、材质的漫反射系数和光线的入射角。

■ 镜面反射 (Specular reflectance)

镜面反射光来自特定方向，并且倾向于沿首选方向从表面反射。从高质量镜子上反射的良好准直激光束产生几乎 100% 的镜面反射。闪亮的金属或塑料具有高镜面反射成分，而粉笔或地毯几乎没有。镜面反射可以被认为是**光泽度**。

这种效果可以在任何有光泽的表面上看到，从表面反射的光往往与光源具有相同的颜色（从红苹果反射的白光中出现的白点）。当视点方向偏离反射方向（等于围绕表面法线镜像的光源方向）时，反射率往往会迅速下降；对于完美反射器（即完美镜子），镜面光仅出现在反射方向上。影响反射光量的因素是材料的镜面反射系数、相对于反射方向的视角和光源强度。



■ 衰减 (Attenuation)

对于理想源，衰减与距离 d 的平方成反比，但我们可以通过使用以下距离衰减模型获得更大的灵活性，

$$f(d) = \frac{1}{k_c + k_l d + k_q d^2}$$

光照模型 (Lighting/illumination/shading model)

光照模型用于计算对象表面上被照亮位置的颜色。

表面渲染方法 (surface rendering method) 使用照明模型的颜色计算来确定场景中所有投影位置的像素颜色。

Phong模型

一个用于快速计算的简单模型

考虑三个组分：

- Diffuse漫反射
- Specular反射
- Ambient环境光

使用4个向量：

- 向光源： \mathbf{l}
- 向观察者： \mathbf{v}
- 表面法线： \mathbf{n}
- Perfect reflector： \mathbf{r}

➤ For each point light source, there are 9 coefficients

- $I_{dr}, I_{dg}, I_{db}, I_{sr}, I_{sg}, I_{sb}, I_{ar}, I_{ag}, I_{ab}$

➤ Material properties match light source properties

- 9 absorption coefficients (漫反射, 反射, 环境光) * (R, G, B) = 9

$$k_{dr}, k_{dg}, k_{db}, k_{sr}, k_{sg}, k_{sb}, k_{ar}, k_{ag}, k_{ab}$$

- Shininess coefficient α 光泽系数

➤ For each light source and each colour component, the Phong model can be written (without the distance terms) as

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

将来自每个点光源的结果加起来即可



多边形着色 (Polygonal Shading)

在顶点着色 (vertex shading) 中，为每个顶点进行着色运算，顶点颜色变为顶点shades，作为顶点属性 (vertex attribute) 发送到顶点着色器 (vertex shader)。

或者可以将参数发送到顶点着色器，并让它计算shade。

默认情况下，顶点着色是平滑着色 (smooth shading) 时会在对象上插值。

也可以用uniform变量对单个shade着色 (常量着色 flat/constant shading)

常量着色

如果三个向量 (l , v , n) 是恒定的，那么每个多边形只需要进行一次着色计算，并且多边形上的每个点都分配相同的阴影。



平滑着色

光栅器 (rasteriser) 对分配给多边形上的顶点的颜色进行插值，每个顶点都有自己的颜色，光栅器可以使用该颜色为每个片段插入shade。

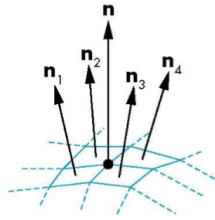
但如果光源较远，或观看者较远，或没有镜面反射，那么平滑着色会用常量对多边形着色。

高洛德着色 (Gouraud Shading)

确定多边形的顶点法线的平均值，并用于使用任何照明模型计算每个顶点的像素强度。

然后通过顶点值的加权平均计算边缘上所有点的强度。

- The vertex normals are determined (average of the normals around a mesh vertex: $\mathbf{n} = (\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 + \mathbf{n}_4) / |\mathbf{n}_1 + \mathbf{n}_2 + \mathbf{n}_3 + \mathbf{n}_4|$) for a polygon and used to calculate the pixel intensities at each vertex, using whatever lighting model.



|| 光源与材质的RGB值

对光源(R,G,B)，材质(R,G,B)到达眼睛的光可计算为 $(R * R, G * G, B * B)$

如果有两个光源，则将分量相加，若和超过1.0定为1。

■ 材质特性

用白色的光照来观察一个材质球，由于rgb都为1，可以看到反射出了材质的颜色。

但若想用纯绿光照亮纯红色的球，那么你看看到将会是黑色的，因为纯红色的材质只会反射红色分量。

材质具有环境反射，漫反射和镜面反射。每种反射与对应光的分量相结合，可认为其中环境反射和漫反射定义了材质的颜色。

漫反射在决定材质颜色上起最重要作用。环境反射影响物体整体颜色。由于环境反射在物体被直接照射的地方最亮，故在物体未被直接光照的地方最明显，对象的总环境反射受受全局环境光和各个光源环境光的影响。漫反射和环境反射不受视点位置影响。

镜面反射会产生高光，反射量取决于视点的位置，反射角方向最亮（如Phong模型）

■ OpenGL：光源与颜色

光源的属性很多，如颜色，位置，方向等

<https://registry.khronos.org/OpenGL-Refpages/gl2.1/xhtml/glLight.xml>

- 指定光源的所有属性：`glLight*()`

```
1 void glLightf( GLenum light,
2               GLenum pname,
3               GLfloat param);
```

light

Specifies a light. The number of lights depends on the implementation, but at least eight lights are supported. They are identified by symbolic names of the form `GL_LIGHT i`, where `i` ranges from 0 to the value of `GL_MAX_LIGHTS - 1`.

pname

Specifies a single-valued light source parameter for light. GL_SPOT_EXPONENT, GL_SPOT_CUTOFF, GL_CONSTANT_ATTENUATION, GL_LINEAR_ATTENUATION, and GL_QUADRATIC_ATTENUATION are accepted.

param

Specifies the value that parameter pname of light source light will be set to.

- `void glLight{if}[v](GLenum light, GLenum pname, TYPEparam)` 创建由 light 指定的光, 可以是 GL_LIGHT0、GL_LIGHT1、... GL_LIGHT7。

所设置的光源的特性由 pname 定义, pname 指定一个命名参数 (参见表格)

TYPEparam 参数指示 pname 特征设置的值;如果使用矢量版本, 则它是指向一组值的指针, 如果使用非矢量版本, 则它是指向值本身的指针。非矢量版本只能用于设置单值光源特性。

参数名称 默认值		意义
GL_AMBIENT	(0.0, 0.0, 0.0, 1.0)	光的环境 RGBA 强度
GL_DIFFUSE	(1.0, 1.0, 1.0, 1.0)	漫反射 RGBA 光强度
GL_SPECULAR	(1.0, 1.0, 1.0, 1.0)	光的镜面反射 RGBA 强度
GL_POSITION	(0.0, 0.0, 1.0, 0.0) (x, y, z, w)	光的位置
GL_SPOT_DIRECTION	(0.0, 0.0, -1.0)	聚光灯方向 (x, y, z)
GL_SPOT_EXPONENT	0.0	Spotlight 指数
GL_SPOT_CUTOFF	180.0	聚光灯截止角
GL_CONSTANT_ATTENUATION	1.0	恒定衰减系数
GL_LINEAR_ATTENUATION	0.0	线性衰减系数
GL_QUADRATIC_ATTENUATION	0.0	二次衰减因子

■ 定向光源和点光源

- 若光源离场景无限远, 则可被认为是平行的光线, 称为定向光源 (directional light source), 比如太阳;
- 若光源靠近场景, 其在场景中的位置决定了影响, 则被称为点光源或位置光源 (positional/point light source), 比如台灯。

定向光源的例子: (w=0)

```
1 GLfloat light_position[] = {1.0, 1.0, 1.0, 0.0};
2 glLightfv(GL_LIGHT0, GL_POSITION, light_position);
```

- 若w=0.0, 则是定向光源, (x,y,z) 描述了其朝向。此方向由model-view matrix转换。

- 若w不为0, 则是位置光源, (x,y,z) 描述了其位置, 由model-view matrix转换并储存在 eye co-ordinates中。

有三种可以实现的效果:

保持固定的光源位置; 围绕静止物体移动的光源; 随视点移动的光源。

聚光灯 (spotlight)

默认情况下, spotlight 功能处于禁用状态, 因为 GL_SPOT_CUTOFF 参数为 180.0。该值意味着光线向各个方向发射 (圆锥体顶点处的角度为 360 度, 因此它根本不是圆锥体)。

GL_SPOT_CUTOFF 的值限制为在 [0.0,90.0] 范围内 (除非它具有特殊值 180.0)。

将 cut-off 参数设置为 45 度: `glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 45.0)`

还可以设置 GL_SPOT_EXPONENT 参数, 来控制光线的集中程度。光强度在圆锥体中心最高。它向圆锥体边缘衰减, 该余弦值是光线方向与从光线到指数点亮的顶点的方向之间的夹角。因此, 较高的光斑指数会导致光源更聚焦。作为定向光源和点光源, 聚光灯方向由model-view matrix转换, 就像它是法线向量一样。

多光源

我们的场景中最多可以有 8 个灯 (可能更多, 具体取决于您的 OpenGL 实现)。

定义点光源和聚光灯:

```
1 GLfloat light1_ambient[] = {0.2, 0.2, 0.2, 1.0};
2 GLfloat light1_diffuse[] = {1.0, 1.0, 1.0, 1.0};
3 GLfloat light1_specular[] = {1.0, 1.0, 1.0, 1.0};
4 GLfloat light1_position[] = {-2.0, 2.0, 1.0, 1.0};
5 GLfloat spot_direction[] = {-1.0, -1.0, 0.0};
6
7 glLightfv(GL_LIGHT1, GL_AMBIENT, light1_ambient);
8 glLightfv(GL_LIGHT1, GL_DIFFUSE, light1_diffuse);
9 glLightfv(GL_LIGHT1, GL_SPECULAR, light1_specular);
10 glLightfv(GL_LIGHT1, GL_POSITION, light1_position);
11 glLightf(GL_LIGHT1, GL_CONSTANT_ATTENUATION, 1.5);
12 glLightf(GL_LIGHT1, GL_LINEAR_ATTENUATION, 0.5);
13 glLightf(GL_LIGHT1, GL_QUADRATIC_ATTENUATION, 0.2);
14
15 glLightf(GL_LIGHT0, GL_SPOT_CUTOFF, 45.0);
16 glLightfv(GL_LIGHT0, GL_SPOT_DIRECTION, spot_direction);
17 glLightf(GL_LIGHT0, GL_SPOT_EXPONENT, 2.0);
18
19 glEnable(GL_LIGHT1);
20 glEnable(GL_LIGHT0)
```

全局环境光 (Global Ambient Light)

每个光源都可以为场景提供环境光。可能还有其他环境光不是来自任何特定来源。（比如多次反射的光，这里模拟近似）

GL_LIGHT_MODEL_AMBIENT 参数用于指定此类全局环境光的 RGBA 强度

```
1 GLfloat lmodel_ambient[] = {0.2, 0.2, 0.2, 1.0};
2 glLightModelfv(GL_LIGHT_MODEL_AMBIENT, lmodel_ambient);
```

在示例中，用于 lmodel_ambient 的值是 GL_LIGHT_MODEL_AMBIENT 的默认值。由于这些数字会产生少量的白色环境光，因此即使我们没有向场景添加特定的光源，我们仍然可以看到场景中的对象。

本地或无限视点 (Local or Infinite Viewpoint)

视点的位置会影响镜面反射的高光计算：

- 本地视点往往会产生更真实的结果，但整体性能会降低，因为必须计算每个顶点的方向。
- 无限视点与场景中任何顶点之间的方向保持不变，因此需要的计算能力更少。

默认情况下假设无限视点，下面是更改为本地视点（到(0, 0, 0)）的方法：

```
1 glLightModeli(GL_LIGHT_MODEL_LOCAL_VIEWER, GL_TRUE)
```

双面照明

我们通常会为正面多边形设置光照条件，而背面的多边形通常没有正确照明。

如果对象是球体，则只会看到正面，因为它们是球体外部的面。因此，在这种情况下，背面的多边形看起来是什么并不重要。但是，如果球体被切掉以便可以看到其内表面，我们可能需要根据定义的照明条件完全照亮内表面；我们可能还希望为背面提供不同的材质描述。

定义材质

类似于定义光源：<https://registry.khronos.org/OpenGL-Refpages/gl2.1/xhtml/glMaterial.xml>

```
1 void glMaterial{if}[v](GLenum face, GLenum pname, TYPEparam)
```


设置的特定材料属性由 pname 标识，其所需值由 TYPEparam 给出，TYPEparam 是指向一组值的指针（如果使用矢量版本）或实际值（如果使用非矢量版本，仅用于设置 GL_SHININESS）。

pname 的可能值显示在表格中。请注意，GL_AMBIENT_AND_DIFFUSE允许将环境光和漫反射材质颜色同时设置为相同的 RGBA 值。

通过为GL_EMISSION指定 RGBA 颜色，我们可以使对象看起来像是发出该颜色的光。此功能可用于模拟场景中的灯和其他光源对象。

face 参数可以是 GL_FRONT、GL_BACK 或 GL_FRONT_AND_BACK，以指示材质应应用于对象的哪个面。

参数名称 默认值		意义
GL_AMBIENT	材质的环境颜色 (0.2, 0.2, 0.2, 1.0)	
GL_DIFFUSE	材质的漫反射色 (0.8, 0.8, 0.8, 1.0)	
GL_AMBIENT_AND_DIFFUSE		材料的环境色和漫反射色
GL_SPECULAR	材质的 (0.0, 0.0, 0.0, 1.0) 镜面反射颜色	
GL_SHININESS	0.0	镜面反射指数
GL_EMISSION	材质的 (0.0, 0.0, 0.0, 1.0) 自发光颜色	
GL_COLOR_INDEXES	(0,1,1)	环境、漫反射和镜面反射颜色指数

■ 总结：

光照和材质都对图形渲染有重要影响，需要考虑一系列因素：光源（位置，定向，类型），照明效果（环境，漫反射，镜面反射），照明模型（如Phong模型），着色方法（flat, smooth, 如 Gouraud)

全局环境光和多个光源 与材料属性相结合。

使用近似来减少昂贵的计算工作。