$$f(m,M,x)=\begin{cases} m+\dfrac{(M-m)x}{60} & x\in[0,60) \\ M & x\in[120,180) \\ m+\dfrac{(M-m)(240-x)}{60} & x\in[180,240) \\ m & x\in[240,360) \end{cases}$$

3.2.3 光照模型

物体能够表现颜色，主要是由于各种光源发出的可见光经物体反射/透射到人的眼睛中所产生的刺激反映，因此，可以说颜色来源于光源，在不同的光照条件下，由于物体对光的吸收能力不同，因而表现出不同的颜色。

计算机图形生成中一般使用四种光源类型来模拟自然界中的光照条件（向世明，1999），它们是：

1) 环境光：又称为泛光，是一种均匀散布的光，用于模拟环境中周围物体散射到物体表面再反射出来的光。
2) 漫反射光：物体反射光中向空间均匀反射的光，反射光与入射光入射角的余弦成正比，光线垂直于物体表面时，物体最亮（反射=入射）。
3) 镜面反射光：反射光向一定的方向发射，在镜面光的照射下，物体表面形成一个非常亮的区域(高光区)，因此亦称为高光。
4) 辐射光：只能照亮光源位置附近的特殊区域的光源，主要用于模拟自发光的物体。

三维模型的光照模型就是根据光学物理的有关规律计算出物体表面各点从光源发出的光投射到观察者眼中所呈现的光的亮度和颜色。常用的光照模型有：朗伯漫反射模型、Phong 反射模型、Cook-Torrance 反射模型、Torrance-Sparrow 光照模型。

（1）朗伯漫反射模型

假定所有物体均为朗伯体，即物体向四面八方均匀反射光线，景物上任一点反射出的光亮度为：

$$I=I_{\alpha}K_{\alpha}+\sum_{i=1}^{N}I_{d}K_{d}\cos\theta_{i}$$

式中，I_{α}、K_{α}为照射在物体上的泛光强度和泛光系数，I_{d}、K_{d}为入射光的强度和物体表面的漫反射系数，θ_{i}为光源入射角，N 为光源数目。

（2）*Phong* 光照模型

考虑镜面光源产生的高光区，将光亮度表示为：

$$I=I_{\alpha}K_{\alpha}+\sum_{i}I_{d}(K_{d}\cos\theta+K_{s}con^{n}\alpha_{i})$$

其中，K_{s}为空间物体表面的镜向反射系数，α_{i}为光源 i 在物体表面产生的

镜面反射与观察方向之间的夹角，N 为镜面反光汇聚指数，与物体表面的光滑度成正比。

（3）*Cook-Torrance* 光照模型

在 *Phong* 光照模型中，物体表面的漫反射系数 I_d 和镜面反射系数 K_s 多被视为常数，根据光学原理，物体对不同波长的入射光有不同的漫反射系数和镜面反射系数，而且，漫反射率还与入射角度有关，为此，*Cook-Torrance* 模型引入双反射率 R 表示物体表面漫反射系数和反射系数与入射光波长和入射角的关系，即：

$$R = K_d R_d + K_s R_s$$

式中 K_d、K_s 为自由参数，与物体表面性质有关，满足 $K_d + K_s = 1$，R_d、R_s 分别为物体表面的漫反射和镜面反射的双向反射率，光的强度表示为：

$$I = K_d I_a R_a + \sum_i I_i (NL_i) d\varpi (K_d R_d + K_s R_S)$$

其中，R_a 为双向泛反射率。

（4）Torrance-Sparrow 光照模型

假定物体表面是由许多细小的、法向随机分布的微平面组成，当光线照射到物体表面时，反射情况由微平面的法向分布情况、反射率、微平面之间的相互遮盖关系决定，物体反射光亮度为：

$$I = K_a I_a + (K_d (N \cdot L) + K_s DFG / (N \cdot E)) I_j$$

式中，D 为微平面方向分布函数，F 为反射率函数，G 为几何衰减因子，L 为光源方向，E 为视线方向。

3.2.4 消隐处理

经过坐标变换、颜色模型和光照设定后，可以在计算机屏幕上显示生成的三维图形，由于计算机绘制是基于二维屏幕坐标，三维物体的深度不可能在屏幕上表示出来，为了生成逼真的三维物体视图，必须在显示前进行物体不可见部分的隐藏，通过计算处理后，不绘制那些实际观察中不可见的部分；如果不进行消隐，则不管两个物体之间的真实的空间位置关系如何，后绘制的物体将覆盖先绘制的物体。常用的消隐算法有画家算法、深度缓存法、光线追踪法。

（1）画家算法

先设置背景颜色，然后把物体的各个面按其距视点的远近进行排序，离视点远的在表头，离视点近的在表尾，构成深度优先级表，绘制时，从表头到表尾依次取出各个面并在绘制区域内显示，从而完成由远及近的物体面的绘制，这一算法与画家作画的过程类似，因此称之为画家算法。

（2）深度缓存算法