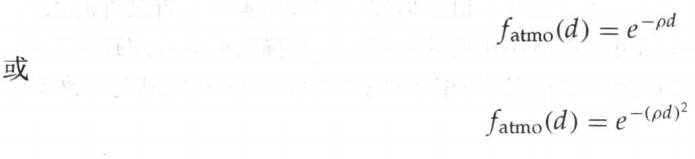
**雾**

在使用雾时，opengl把一种指定的雾颜色与完成了所有其他颜色计算的几何图形进行混合

对象颜色上的雾气效果是光照模型中有时会考虑的另一因素。雾气使颜色变淡、使对象变模糊，因此我们可以按照模拟空气的灰尘、烟或雾的多少来指定一个函数秀该表面颜色。雾气效果常用指数衰减函数来模拟：



赋给d的值是从观察位置到对象的距离，我们还在这两个指数函数中使用参数p来为雾气设定一个正密度。较高的p值生成较稠密的雾气且导致表面更柔和。在以对象的表面颜色计算出来后，将该颜色与雾气函数之一相乘来减少其强度，减少量依赖于雾气密度设定的值。

我们也可以使用线性的深度提示函数来简化雾气衰减函数从而代替指数函数。这使远距离对象的表面颜色强度减弱，但这样就失去了改变雾气密度的控制。

我们有时要模拟雾气的颜色。例如，有烟房间的空气可以用蓝灰色或用淡灰色建模，混合雾气颜色和对象颜色的计算如下：



这里的是一个指数或线性雾气衰减函数

即最终的颜色=对象颜色\*衰减指数+雾颜色\*(1-衰减指数)

float fogColor[] = {1.0f, 1.0f, 1.0f, 1.0f};

glEnable(GL\_FOG); // 开启雾

glFogfv(GL\_FOG\_COLOR, fogColor); // 设置雾颜色

glFogf(GL\_FOG\_START, 0.0f); // 雾从多远开始

glFogf(GL\_FOG\_END, 50.0f); // 雾从多远停止

glFogi(GL\_FOG\_MODE, GL\_LINEAR); // 采用哪种方程式

glFogf(GL\_FOG\_DENSITY, 0.5f); // 雾的稠密度

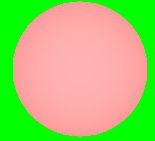
glPushMatrix();

glTranslatef(0.0f, 0.0f, -30.0f);

glColor4f(1.0f, 0.0f, 0.0f, 1.0f);

glutSolidSphere(5.0f, 50, 50);

glPopMatrix();



我们看到的球体边缘的像素点是(255,153,153)

实际上衰减因子=(50-30)/(50-0)=0.4

Final color = (1.0f, 0.0f, 0.0f) \* 0.4f + (1.0f, 1.0f, 1.0f) \* 0.6f

= (1.0f, 0.6f, 0.6f) = (255, 153, 153)

雾模式 雾方程式

GL\_LINEAR f = (end - c) / (end - start)

GL\_EXP f = exp(-d \* c)

GL\_EXP2 f = exp(-(d \* c) ^ 2)

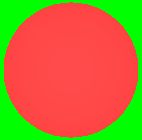
其中d表示稠密度，c表示距离

也就是说对于GL\_LINEAR而言，稠密度对最终颜色没有任何影响

但是对于GL\_EXP与GL\_EXP2而言，抽密度越大，衰减因子就越小，物体颜色所占比例就越小

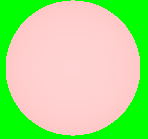
glFogi(GL\_FOG\_MODE, GL\_EXP); // 采用哪种方程式

glFogf(GL\_FOG\_DENSITY, **0.01f**); // 雾的稠密度



glFogi(GL\_FOG\_MODE, GL\_EXP); // 采用哪种方程式

glFogf(GL\_FOG\_DENSITY, **0.05f**); // 雾的稠密度



注意：GL\_FOG\_START和GL\_FOG\_END仅对GL\_LINEAR雾有效，GL\_FOG\_DENSITY对于GL\_LINEAR雾无效

片断与眼睛的距离可以通过两种方法进行计算。有些实现（以NVIDIA硬件著称）将使用实际的片断深度。其他实现（多见于ATM芯片）使用顶点与顶点之间的插值。前者有时称为片断雾，后者有时称为顶点雾。片断雾比顶点雾更为复杂，但它长航具有更好的质量

雾坐标

我们并不一样要让opengl计算雾距离，也可以自己完成这种计算，这种值称为雾坐标。可以采用glFogCoordf函数手动设置。除非用下面这个函数更改了雾坐标的来源，否则雾坐标是被忽略的

glFogi(GL\_FOG\_COORD\_SRC, GL\_FOG\_COORD)