# 17.4 模拟反射

如前所述，环境贴图在实现天空纹理映射时可以达到比较好的效果。环境贴图的另一个主要用途是为任意物体模拟反射（该技术只能反射环境贴图中的图像）。图17.7说明了如何使用环境贴图来实现镜面反射。这个表面就像是一面镜子：观察点e可以看到由p点反射回来的环境图像。

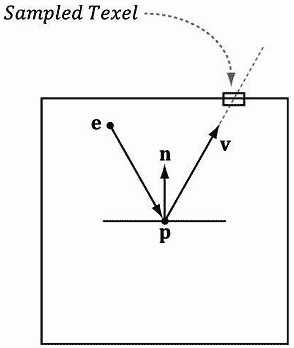


图17.7 这里**e**是观察点，**n**是点**p**的表面法线，**v**是**ep**向量的反射向量。我们可以通过反射向量**v**将纹理元素映射到表面点**p**上（即，我们可以把v作为查找向量）。通过一方式，观察点就可以看到反射后的环境图像了。

我们为每个像素计算反射向量，然后用它来对环境贴图进行采样：

litColor = texColor\*(ambient + diffuse) + spec;

if(gReflectionEnabled)

{

float3 incident = -toEye;

float3 reflectionVector = reflect(incident, pin.NormalW);

float4 reflectionColor = gCubeMap.Sample(samAnisotropic,refle ctionVector);

litColor += gMaterial.Reflect\*reflectionColor;

}

通常，像素的颜色不完全取决于反射颜色（除非镜子的反射率是100%）。所以，我们必须修改一下光照方程，加入一个反射项**m**R⊗**c**R。这里**c**R是环境贴图的采样颜色，**m**R是应用程序指定的材质颜色，它决定了表面对**c**R的反射数量。例如，当表面只反射红光时，你应该将**m**R设为(1,0,0)，使表面只从环境贴图中反射红光。我们之前定义的**Material**结构已经包含了一个反射属性，本章终于可以派上用场了：

struct Material

{

float4 Ambient;

float4 Diffuse;

float4 Specular;

float4 Reflect;

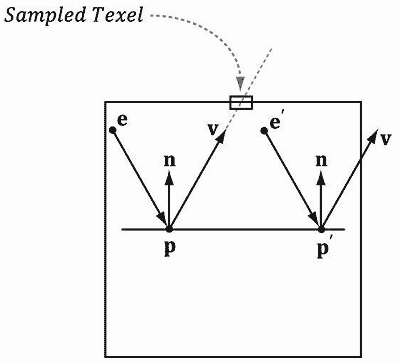
};

现在存在的一个问题是在光照方程引入附加的反射项后，物体颜色显得过于饱和。由于反射项的添加，每个像素颜色值都会增大，物体表面的亮度会显得过高。本质上，如果我们要将一个额外的反射颜色添加进来，那么就必须从其他项中减去一些颜色，以求得平衡。一般可以通过降低环境材质系数和漫反射材质系数来实现，使表面反射的环境光和漫反射光更少一些。另一种方式是计算环境贴图的采样颜色**c**R和普通光照颜色**s**之间的加权平均值：

**f** = *t***c**R + (1 − *t* )**s** 其中0≤*t*≤1

通过一方式，我们给环境贴图的采样颜色添加了一个权值*t*，从普通光照颜色中减去了等量的颜色，以保持平衡。这里的参数*t*可用来控制表面的反射率。

图17.8说明通过环境贴图映射无法达到令人满意的平面反射效果。这是因为反射向量无法表达完整的信息（它不包含坐标位置）；我们实际需要的是一条反射射线，并且要让该射线与环境贴图相交。射线具有位置和方向，而向量只具有方向。我们可以从图中看到，两条反射射线**r**(*t*) = **p** + *t***v**和**rʹ**(*t*) = **pʹ** + *t***v**与不同的立方体贴图元素相交，所以应该得到不同的颜色。不过，由于两条射线具有相同的方向向量**v**，而方向向量**v**又被唯一地用于查找立方体贴图，所以当观察点位于**e**和**e**ʹ时，**p**和**p**ʹ会映射到同一个纹理元素上。对于平面物体来说，这是环境贴图映射的一个严重缺陷。对于曲面来说，环境贴图映射的这一缺陷并不明显，因为曲面会产生反射向量的变化。[Brennan02]在该问题上给出了一个基本可行的解决方案。

****

**图17.8 当视点位于e和eʹ时，反射向量v分别对应于两个不同点p和pʹ。**