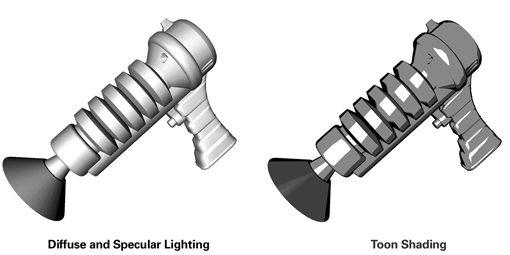
# 非真实感渲染 Nonphotorealistic Rendering

## 简介

虽然大多数时候开发者会尝试不断提高计算机生成图像的真实感，然而有时我们也会试图制作看起来不真实的图像，比如在设计程序中绘制线框图，或绘制卡通图像。我们把这些例子称为非真实感渲染，简称NPR。

## 卡通着色 Toon Shading



上图展示了普通着色方式与卡通着色方式的对比。在卡通着色中，镭射枪只有3种色调：

* 明亮的漫射区域
* 黑暗的漫射区域
* 镜面反射的高光区域

## 实现卡通着色 Implementing Toon Shading

### 要点

卡通着色与普通着色方式最大的区别便是尽量减少阴影的变化。

卡通着色最重要的三个组成部分是：

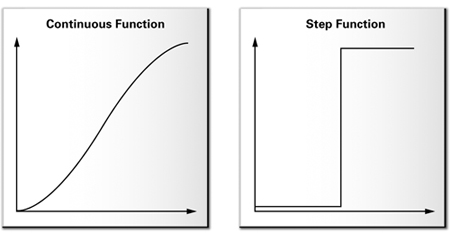
#### 漫射着色只需要两个值来表现：一个对应亮区，一个对应暗区。

#### 镜面高光只需要使用一种色彩

#### 对象需要有轮廓线

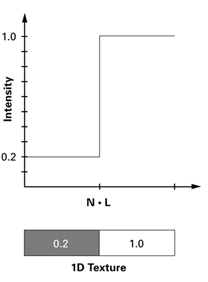
### 漫射着色 Diffuse Shading

卡通着色器需要将漫射光转化为个别几种颜色。在普通着色方式中，漫射光在物体表面的变化通常是逐渐从0变化到1，一种实现卡通着色器的方式便是将这种连续变化修改成2个离散的值，比如将0到0.5之间的值全部修改为0.2，0.5到1之间的值全部修改为1。在数学上，我们称后者为阶梯函数Step Function。



基于这一原理，卡通着色器的漫射部分使用N · L（法线单位向量与光线单位向量的点积）作为分辨“亮区”与“暗区”的参数。

将N · L的值映射到阶梯函数可以使用纹理查找的方式进行。类似于使用标准化立方体贴图实现“标准化”函数，我们可以使用一个1D纹理来实现阶梯函数。在这个方案中，由于N · L是一个标量，因此使用1D纹理便足够了。



### 镜面高光 Specular Highlights

与处理漫射着色类似，在卡通着色中，镜面高光没有逐渐的强度变化，只有存在与不存在，因此也可以使用一个1D纹理来实现。

### 轮廓勾勒 Silhouette Outlining

在卡通着色中，还需要勾勒出对象的轮廓。找到对象轮廓的一个简单方法是使用N · V（法线单位向量与视线单位向量的点积）作为量尺。当N · V的值接近于0时，表示它处于一个对象的轮廓线上。同样的，我们可以使用一个1D纹理实现轮廓线取值。

### 解决锯齿 Addressing Aliasing

使用上述方法实现的卡通着色在绘制动画时容易出现图像闪烁现象（由锯齿导致）。产生这种现象的原因是漫射、镜面与边缘的检测都太严格了，导致对象移动时会产生大幅的波动。这是使用阶梯函数的自然结果——当自变量发生轻微变化时，因变量可能会产生突变。

解决这个问题的方法是使用一个更大的1D纹理来存储一个比较平滑的过渡。

### 技术缺陷 Problems with This Technique

上述方法实现的卡通着色还有可能导致一个问题：当绘制具有锐利边缘的对象时，N · V会在边界发生突变，使轮廓完全变成黑色。

解决这个问题的一个方法是将一个对象绘制两次：第一次将对象沿顶点法线稍微扩大，并绘制成黑色，第二次正常绘制对象（忽略轮廓的绘制）。但这个方法也有一个缺陷：只能解决边缘轮廓，而无法绘制内部边缘。