# 基础篇

* 渲染流水线

三阶段：应用、几何、光栅化

应用：CPU负责

几何、光栅化：GPU负责

几何：顶点着色器（可编）-曲面细分着色器（可选）-几何着色器（可选）-裁剪-屏幕映射

光栅化：三角形设置（固定）-三角形遍历（固定）-片元着色器（可编）-逐片元操作

逐片元：片元-模板测试-深度测试-混合-颜色缓冲区

* UnityShader 基础

ShaderLab

* 数学基础

坐标系

点和矢量

矩阵

变换

坐标空间

法线变换

# 初级篇

* 基础光照

标准光照模型-漫反射、高光反射

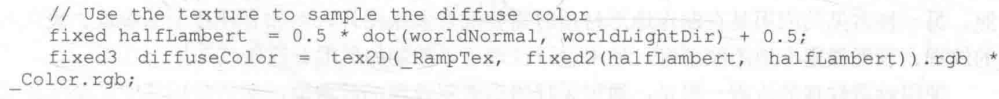
* 基础纹理

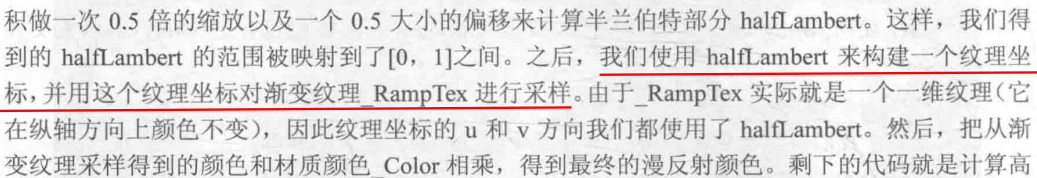
单纹理-纹理属性，tex2D函数：通过UV对纹理采样，UV即位置，即拿到纹理图上对应位置的纹素

凹凸映射-高度纹理、法线纹理

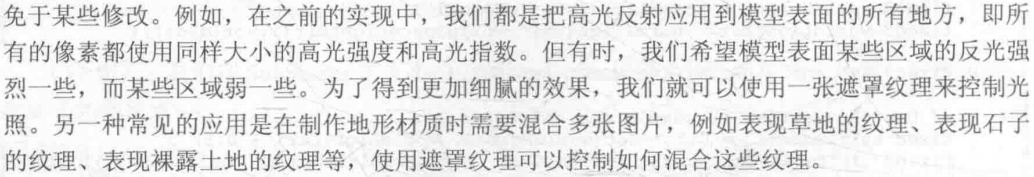
渐变纹理-

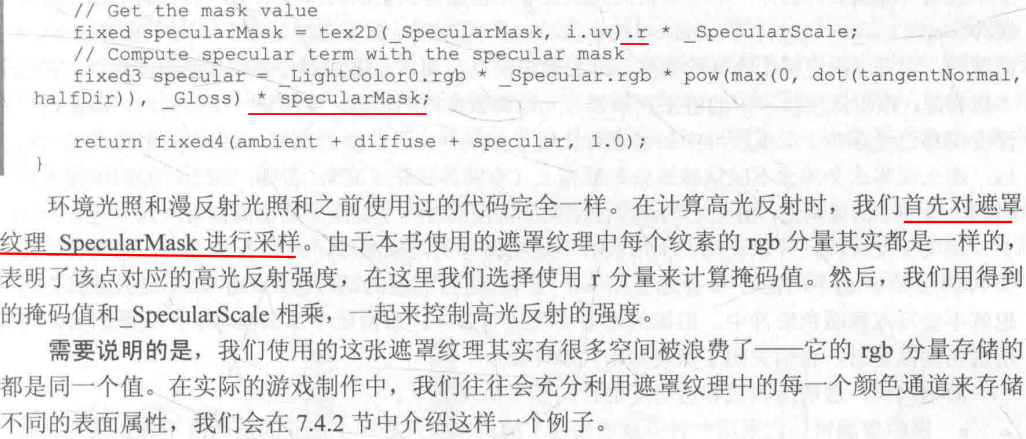
常用：控制漫反射光照结果，halfLambert值越大说明越亮，渐变纹理是一维纹理，也是从暗到明排列，所以直接进行二维采样





遮罩纹理-



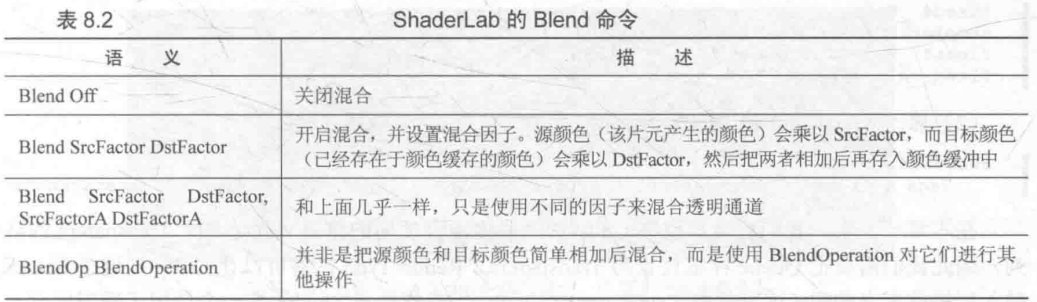


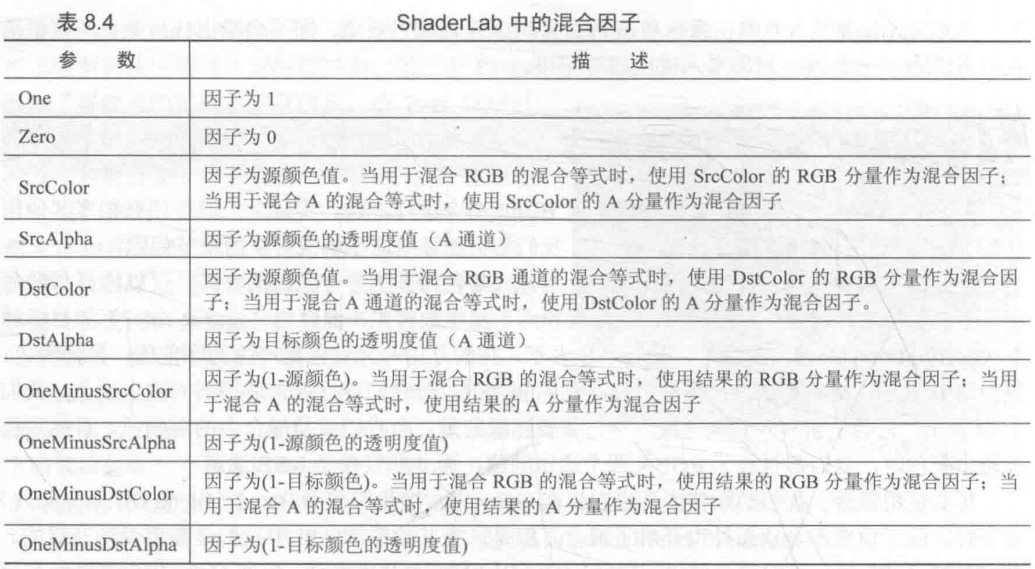
* 透明效果

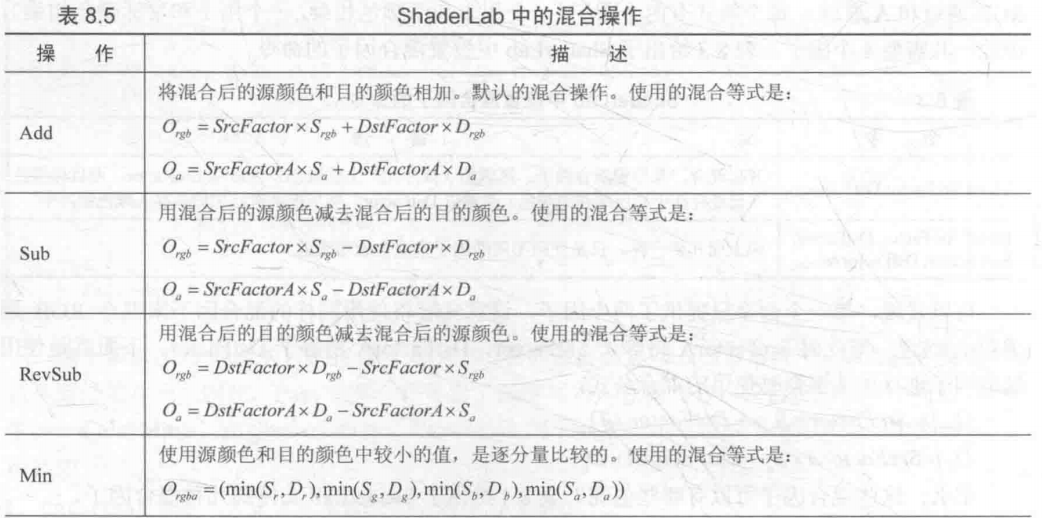
透明度测试

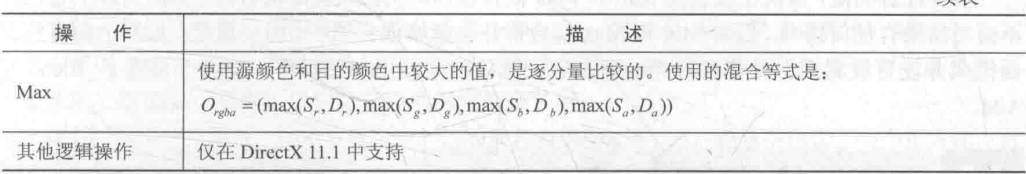
透明度混合

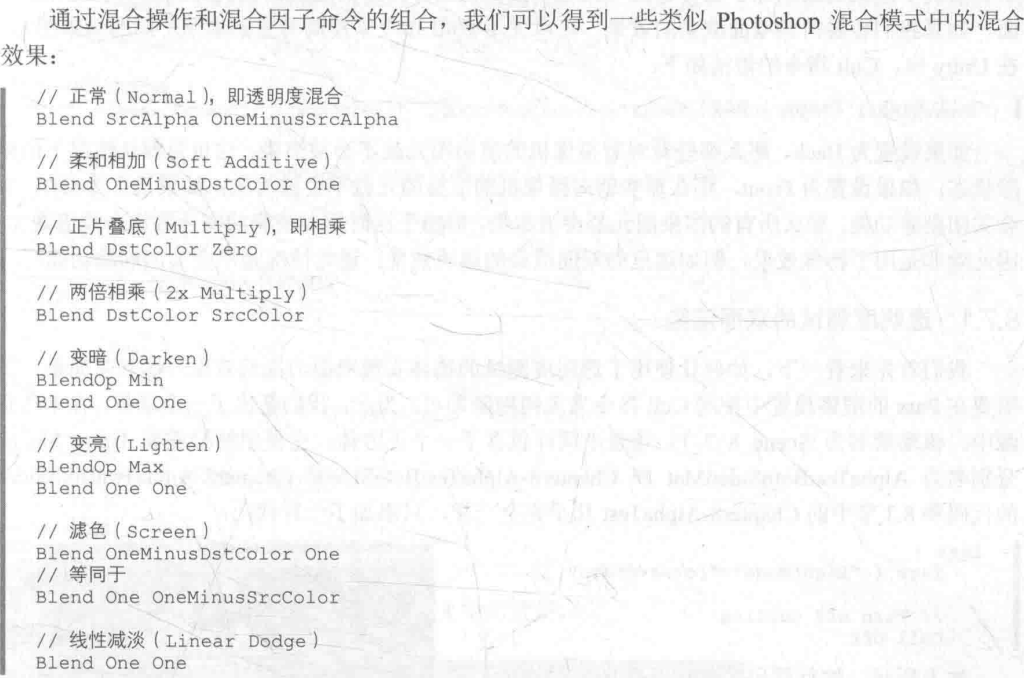








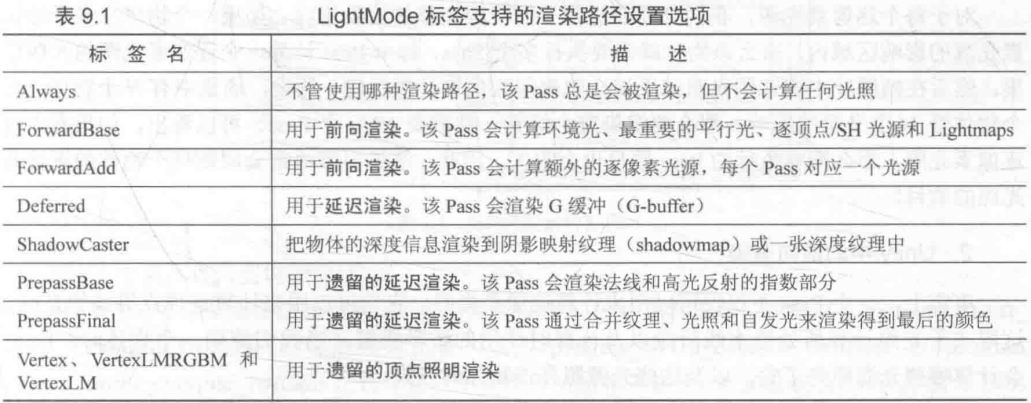




# 中级篇

* 复杂光照

渲染路径：前向、延迟、顶点照明（已抛弃）



**前向：**

缓冲：颜色、深度

3中处理光照方式：逐顶点、逐像素、球谐

两种Pass：Base Pass & Additional Pass

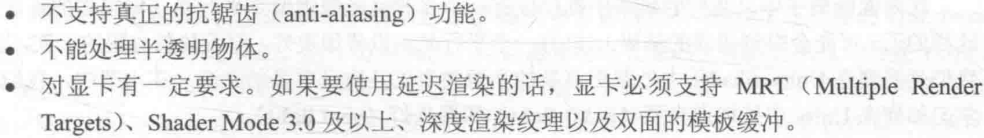


**延迟：**

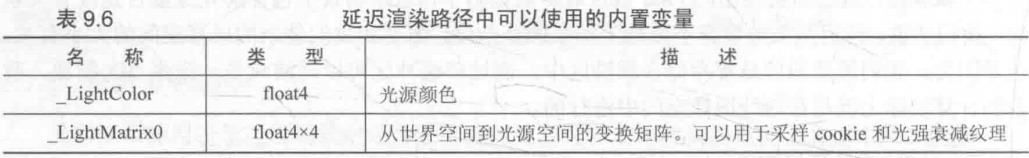
缓冲：颜色、深度、G

古老的方法，但是前向达到瓶颈后，又开始流行

缺点：

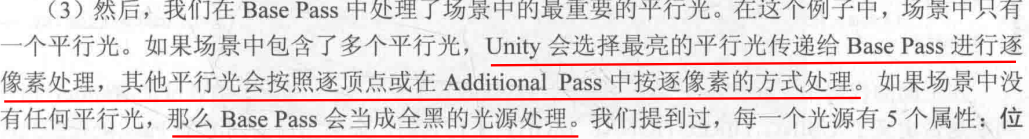


Pass：第一个G缓冲 第二个真正的光照模型

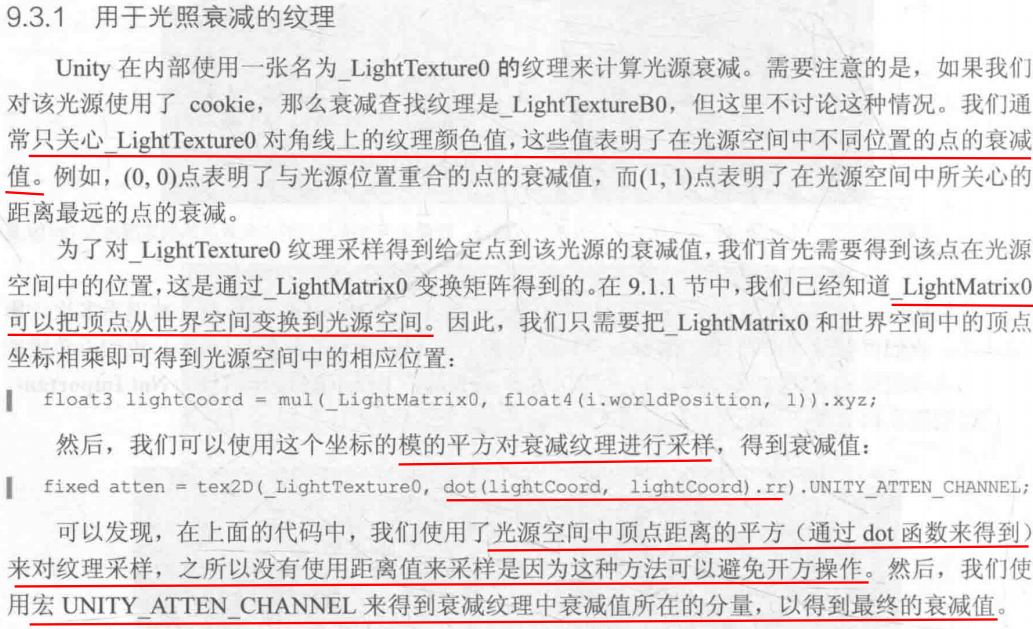


**光源：**

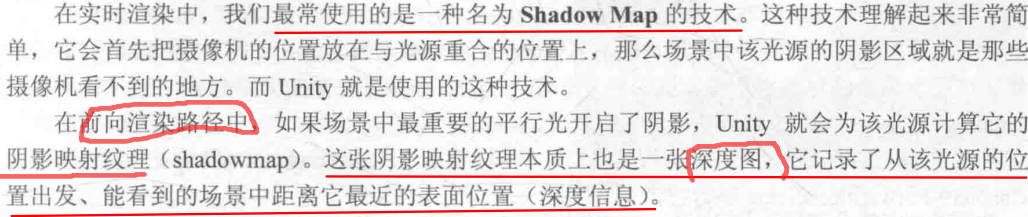
**光源属性**：位置、方向、颜色、强度、衰减

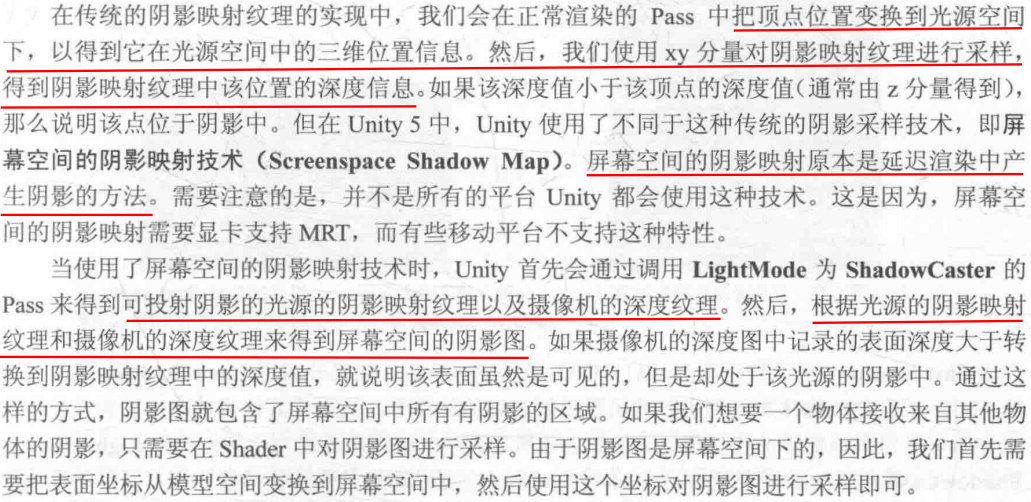


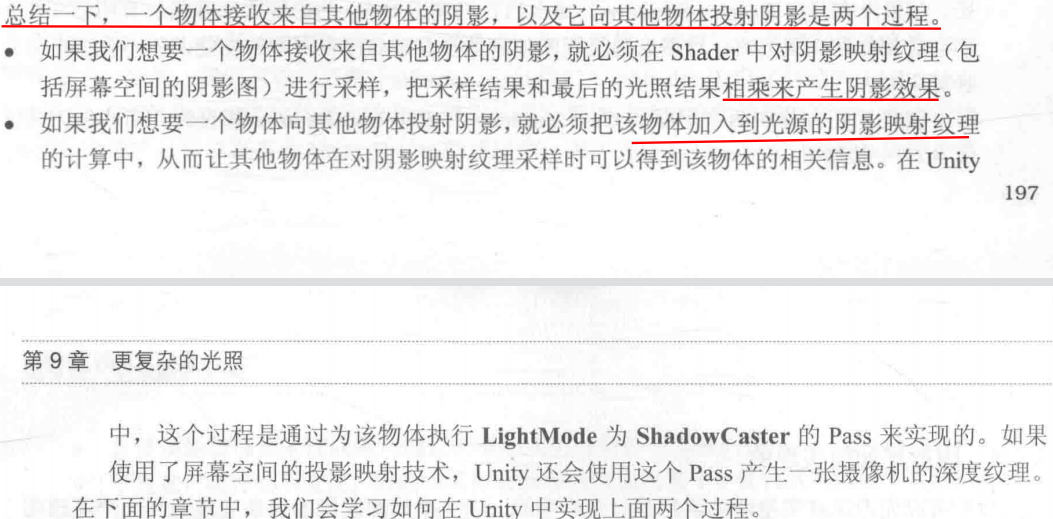
**光照衰减**



**阴影：**

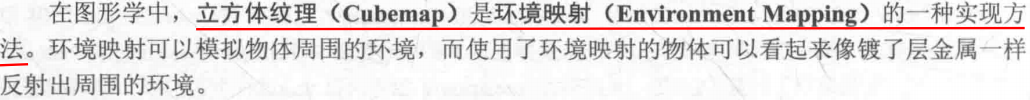






* 高级纹理

**立方体纹理**

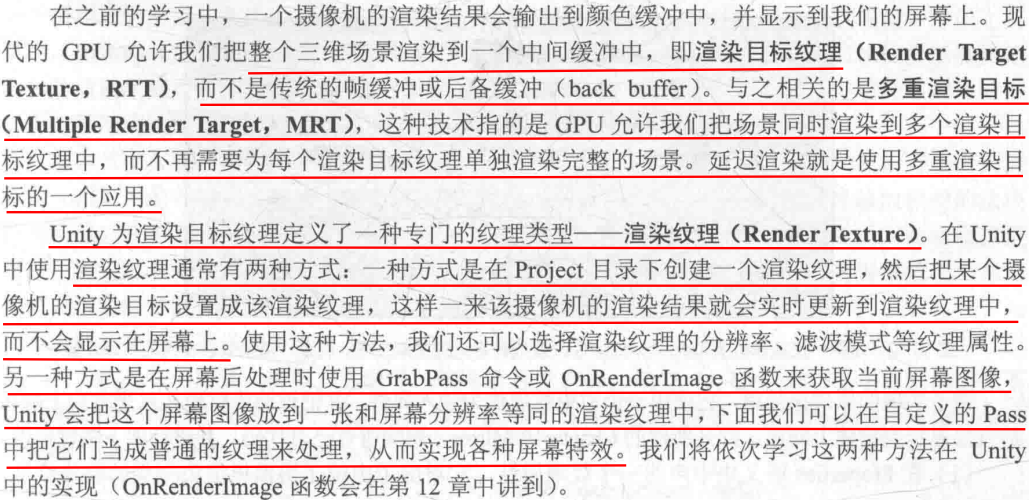


反射

折射

菲涅尔反射

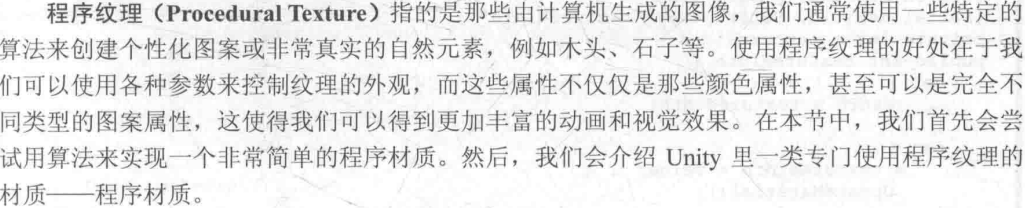
**渲染纹理**



镜子效果

玻璃效果

程序纹理



* 着色器动画



纹理动画---序列帧动画

顶点动画---河流、广告牌

# 高级篇

* 屏幕后处理

边缘检测

高斯模糊

Bloom

运动模糊

* 深度/法线纹理

运动模糊

全局雾效

边缘检测

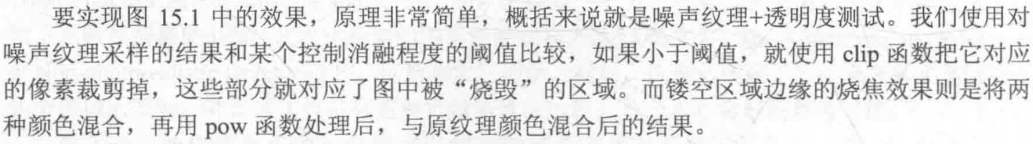
* 非真实感渲染NPR

卡通风格

素描风格

* 噪声

消融



水波

全局雾效

* 优化技术

减少draw call

动态批处理

静态批处理

共享材质

减少需处理的顶点数目

优化几何体

模型LOD技术

遮挡剔除

减少需处理的片元数目

绘制顺序

透明物体

减少实时光和阴影

节省带宽

纹理大小

分辨率缩放

减少计算复杂度

Shader LOD技术

代码优化

根据硬件缩放

# 扩展篇

* 基于物理渲染PBR