

一、原理简述

1、论文原理概述

核心部分的素描渲染效果参考经典论文Real-Time Hatching。（论文链接 <http://hhoppe.com/hatching.pdf>）

以论文中给出的一组Tonal Art Maps为基础，用混合纹理的形式模拟素描笔触效果。

纹理由左到右表示亮度由亮到暗的素描效果，亮的纹理图片是暗的纹理图片的子集。

根据点受到的光照强度采样对应的纹理贴图，6张基础纹理将亮度分为6段，每段混合相邻两张纹理。

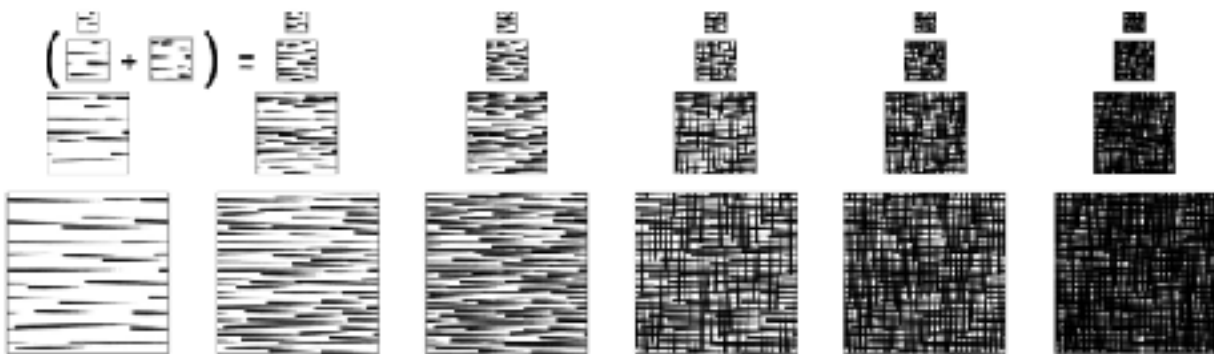


Figure 2: A Tonal Art Map. Strokes in one image appear in all the images to the right and down from it.

2、论文效果



二、实现方法

1、采用three.js实现，主要步骤：（代码见demo.html，有注释）

- 模型、场景构建，使用了three.js提供的几何体搭建简单场景。
- 着色器的实现，主要在片元着色器中实现对应混合纹理贴图。
- 监听鼠标操作，场景移动、缩放。

2、着色器shader的实现方法：

通过光照的强弱，用不同深浅的纹理，实现一种类似素描效果。通过 $N \cdot L$ （光向量 L 与入射点的切面法向量 N ）计算点受光的强弱，将该值分成不同的段，不同的光照强度对应不同深浅的素描贴图

- Phong 光照模型

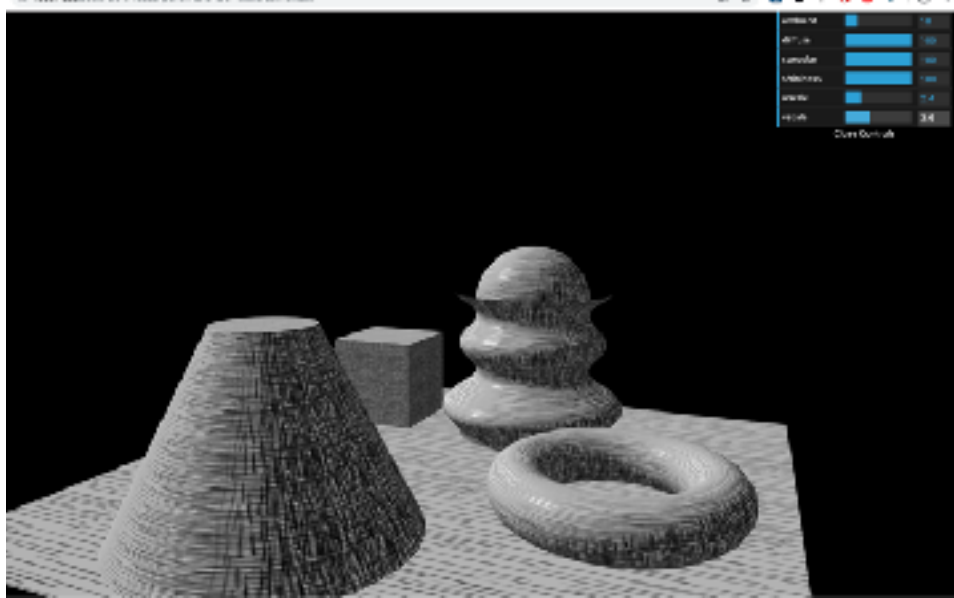
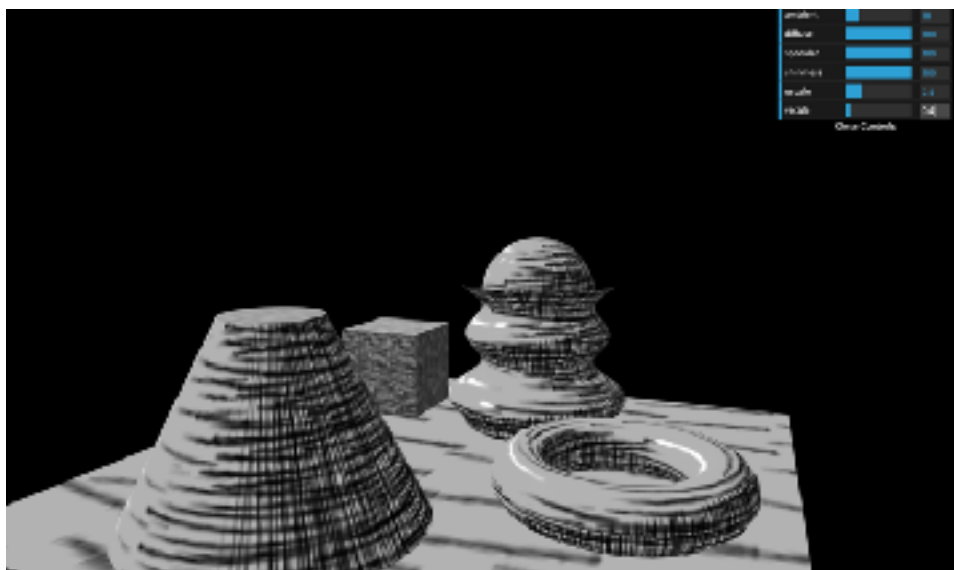
- 环境光照 Ambient Lighting，漫反射光照 Diffuse Lighting，镜面光照 Specular Lighting，光泽度 Shininess

- 纹理坐标(u,v)

- $u = u * uscale$, $v = v * vscale$ ，每个分量乘个线性的值（稀疏、密集）

三、实现效果

调节光照参数对素描笔触效果影响较大，调节uv纹理坐标使得笔触变得稀疏或密集，对于不同形状的物体uv纹理坐标乘以不同的参数可能会得到更好的效果。



四、运行方式

Python 服务器

如果你安装了 Python，在你的工作目录下运行以下命令行：

```
//Python 2.x  
python -m SimpleHTTPServer
```

```
//Python 3.x  
python -m http.server
```

会从当前目录转到 localhost 的 80 端口发起服务，地址栏是这样：

<http://localhost:8000/>

在地址栏输入 http://localhost:8000，找到 demo.html 打开

具体可参考：<https://segmentfault.com/a/1190000011858935>