

# 计算机图形学

## 第八次实验报告

### Shader

PB20000264

韩昊羽

# 一．实验要求

- 配置编译 UGL 框架
- 完成第一个任务：添加置换贴图和法向贴图
- 完成第二个任务：降噪
- （选做）完成第三个任务：点光源阴影

# 二．操作环境

IDE：Microsoft Visual Studio 2019 community

CMake：3.23.1

UGL 及其依赖、UANN

# 三．架构设计

和框架保持一致，未进行修改。

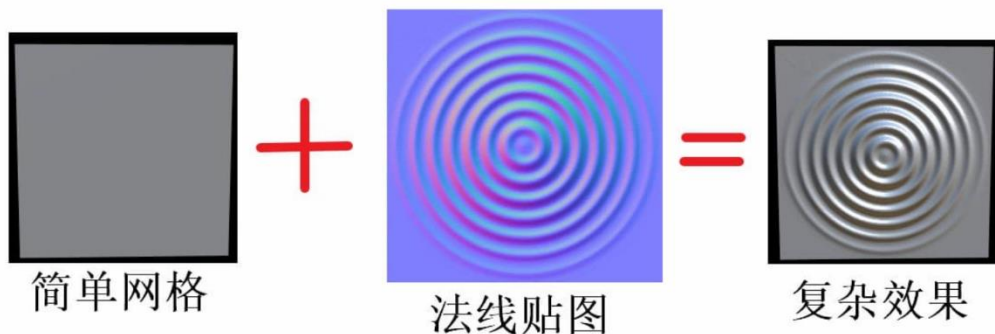
# 四．功能实现

## 4.1 置换贴图和法向贴图

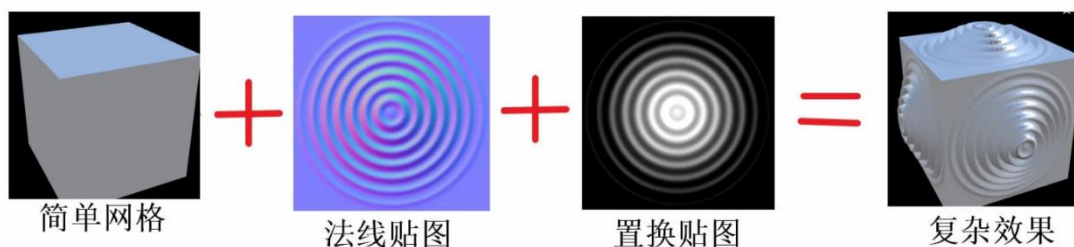
我们在处理物品表面的时候，经常会发现一个面元里的纹理没有凹凸变化，真实感不足。当我们从光的角度看待这个问题时，发现被渲染时内容只跟法向有关。这样我们就可以通过只改变法线分量，再和表面联系起来就可以生成一张真实感很强的照片。法线分量的信息存储在法相贴图中，使用的时候从贴图中读取 RGB 信息转化成法向信息，再对片元的法向进行改变。

$$Normal_{new} = Normal_{old} + TBN * (Normal_{tex} * 2.0 - 1.0)$$

其中从 normalmap 中读取的 normal\_tex 因为在[0,1]间变化，需要调整到[-1,1]范围。TBN 矩阵是因为存在法向贴图中的数据是从片元系得到的，需要转化到世界坐标系。



但有时我们不止希望看起来真实，也希望物体的形状能真正变得凹凸。于是我们引入了置换贴图，即保存每个顶点沿法向偏移的数据量，这样存储一张图片只需要存储黑白数据，存储量少，好提取。在本项目中，由于只涉及到图案的突出，所以直接使用[0,1]范围就行。



$$aPos_{new} = aPos_{old} + Normal * \lambda * displacement$$

## 4.2 去噪

对于有噪音的物体，我们对那些比较尖锐的顶点“抹平”，即想要让顶点向物体的内部偏移，采用 laplace 坐标来实现。即

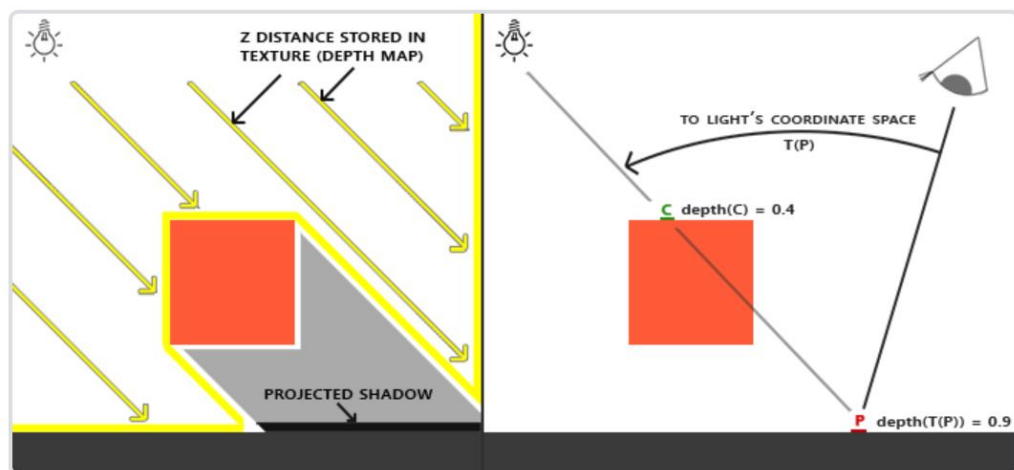
$$\delta_i = p_i - \frac{1}{|N(i)|} \sum_{j \in N(i)} p_j$$

投影到法向上得到  $\bar{\delta}_i = \langle \delta_i, n_i \rangle n_i$ ，然后对每一个顶点进行偏移  $\bar{p}_i = p_i - \lambda \bar{\delta}_i$

注意因为要存在 displacementmap 中，所以要正规化到[0,1]范围，同时引入 bias 和 scale 用来恢复原始值。操作的时候从置换贴图中读取数据即可。

## 4.3 点光源阴影

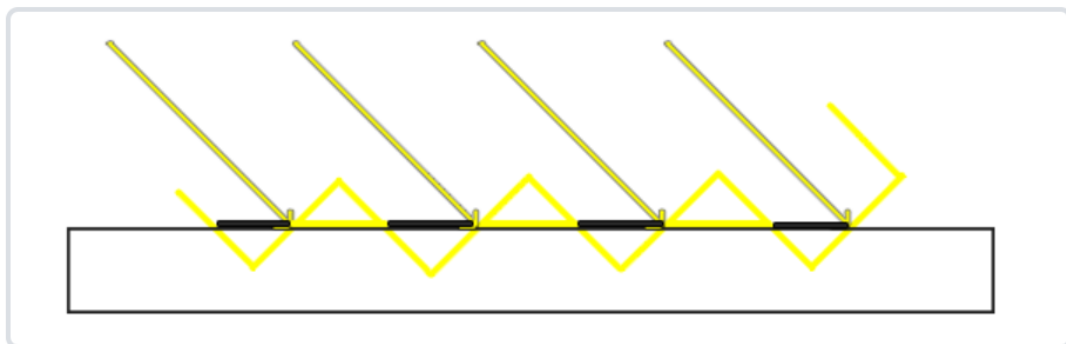
对于物体身上的阴影和明暗变化我们已经处理的很好了。但物体之间的阴影还处理的不好，这是因为我们默认每个物体都在一个没有其他物体的空间里接受光照。所以我们需要一个标记哪些点被遮挡住没有照亮的信息。我们采用阴影映射来解决，思路是从光源的角度渲染镜头，按照在这个空间最近的点的深度存储深度贴图，使用的时候访问目标点的深度和最近点的深度决定是否被照亮。



第一步：生成深度贴图，需要修稿 projection 和 view 矩阵，改成从光源看成场景中，视角和光源照射的一样，然后渲染物体，根据深度生成 map。

第二步：传入矩阵。需要传入摄像机空间的矩阵，可以让点从世界坐标变换到光源坐标。

第三步：判断阴影。根据目标点和最近点的深度来判断是否是阴影，注意这里因为深度贴图是像素形式，所以对于同一个像素点覆盖的区域，如果光线不垂直，会导致有的区域判定在平面上，有的区域不在，解决方法是增加一个微小的误差即可。



## 五．难点

### 5.1 TBN 矩阵的计算

在计算 TBN 矩阵的时候，由 N,T,B 三个相互垂直的分量组成，N,T 向量由框架给出，但 B 需要自己按照向量叉乘算出来，同时也要注意叉乘方向。

### 5.2 邻接顶点的获取

因为在录入 obj 文件的时候没有录入边的信息，只有面的信息，所以需要自己获取邻接矩阵。我选择在负责录入的 SimpleLoader 命名空间中加入了从面信息转化成边信息的代码，也

加入了 edge 变量作为邻接矩阵。这样可以直接访问邻接顶点  $N(i)$

## 5.3 ANN 库的配置

配置 ANN 库的时候需要自己修改 Cmakelist, 但我修改后还是无法正确 include, 不生成 ANN.h 文件, 最后采用了导入 ANN.lib 和 ANN.dll, 但需要在 bulid 之前保留 dll 文件。

## 5.4 接缝问题

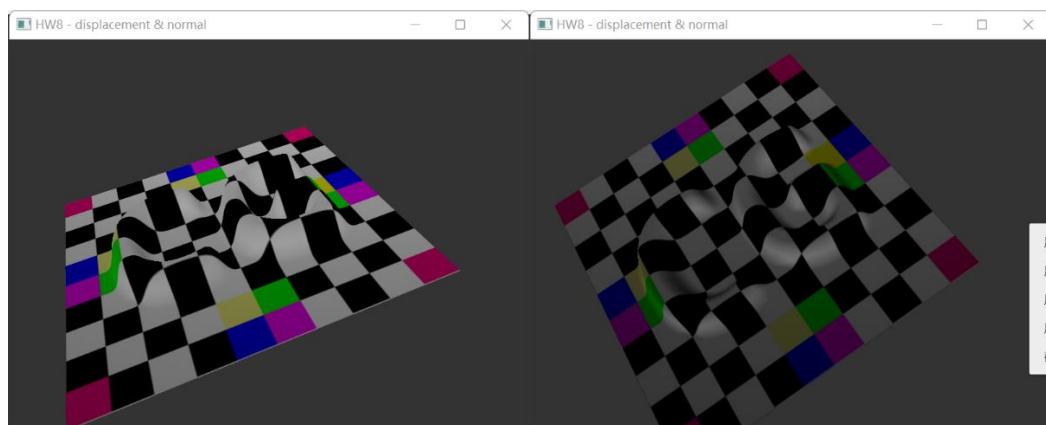
发现去噪之后物体表面出现了许多条裂缝, 是因为原物体中存在 position、normal 相同但邻接点和 texcoords 不同的顶点, 所以在计算 laplace 坐标时不同, 偏移量不同。需要在初始化的时候根据初始坐标找出重合的那些点, 在偏移的时候采取相同的偏移变量。

## 5.5 projection 和 view 自定义

在处理点光源阴影时, 参考资料采用的是 openGL 里的函数, 具体实施则是需要 UGL 框架里的函数, 需要寻找对应的函数并比对参数。有些东西框架已经写好了, 有些没写好也需要自己判断。

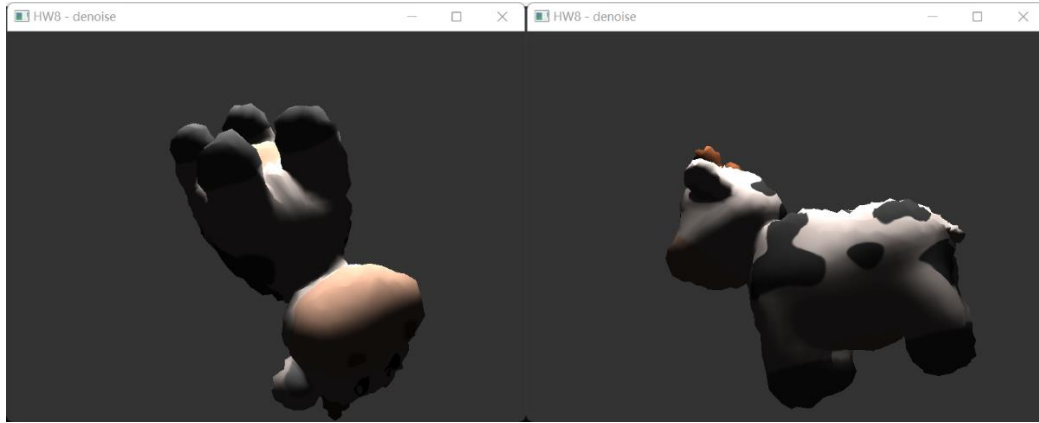
# 六 . 实验结果

## 6.1 置换贴图和法向贴图

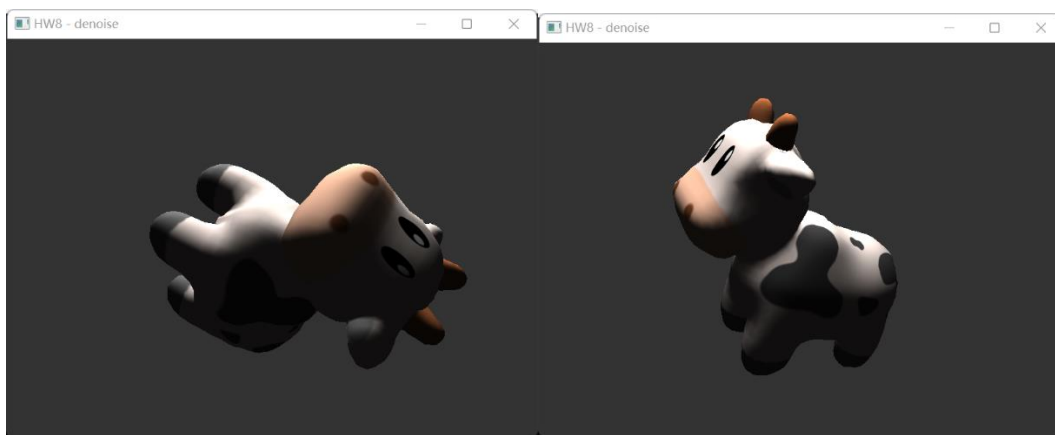


## 6.2 去噪

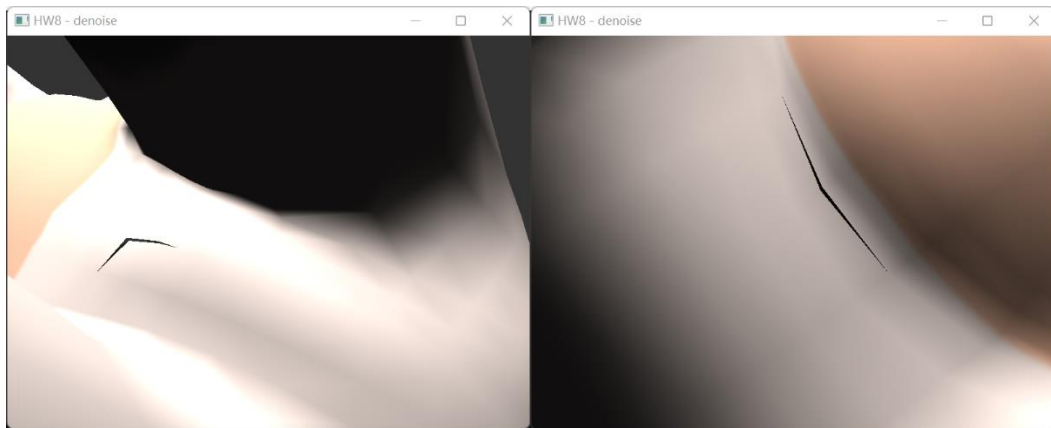
有噪声:



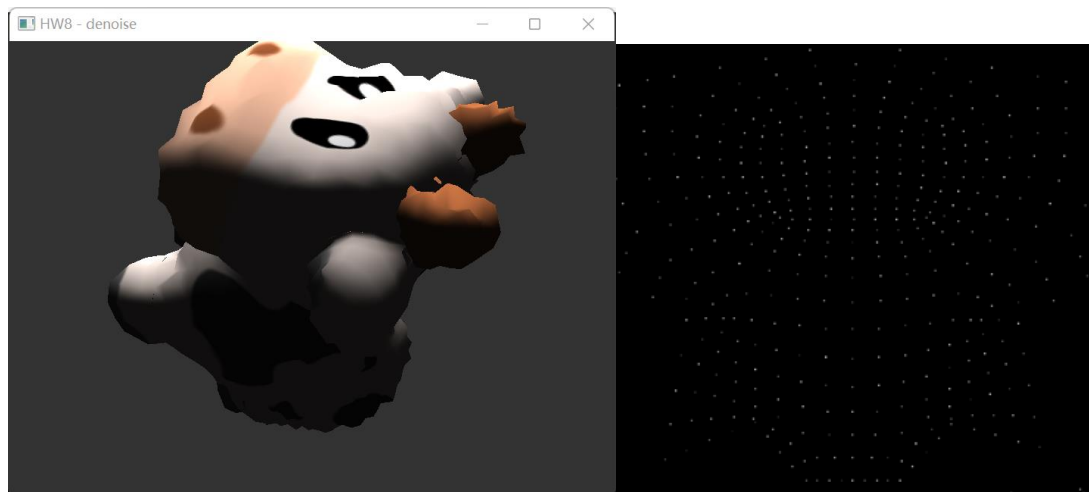
无噪声:



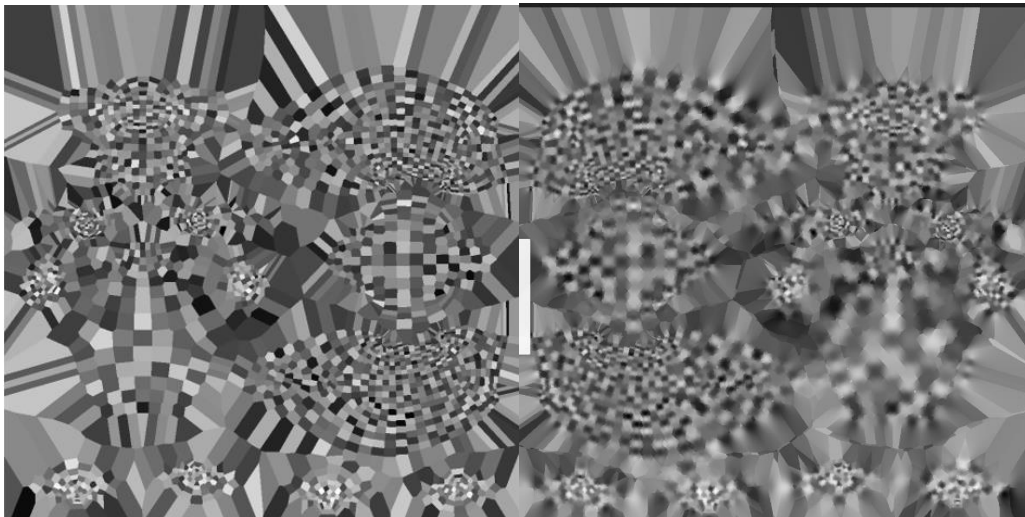
微小的接缝:



不使用最近邻:



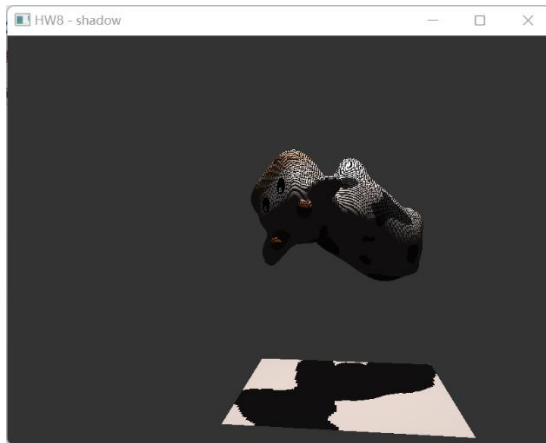
生成的置换贴图：



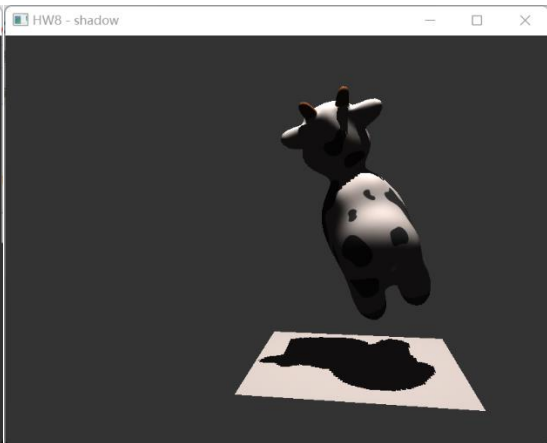
一开始我觉得存入置换贴图的时候是按照纹理坐标来存的，那读取的时候应该还是读的写入的那个，所以不需要最近邻，后来发现着色器读取的方式使插值读取，如果只修改单点的值很可能两边标准不一致而读取到错误的值，所以后来使用 ANN 解决。对于仍存在的接缝问题，大概是存入的时候和读取的时候采用了不同的方式，所以插值出来的值会有细小的偏差。这个问题大概率无法解决，除非我们直接把偏移传入着色器。

### 6.3 点光源阴影

未加阴影偏移：



添加偏移：



总体效果好了很多，但还是有地方存在条纹。

## 七．问题与展望

### 8.1 未解决的问题

- 去噪时候的接缝问题
- 阴影的偏移会导致的误差
- 深度贴图外全部是阴影

### 8.2 future work

- 改进接缝
- 将光源标记出来，添加交互使得可以调整光源
- 使用 PCF 减少锐化
- 改进阴影偏移