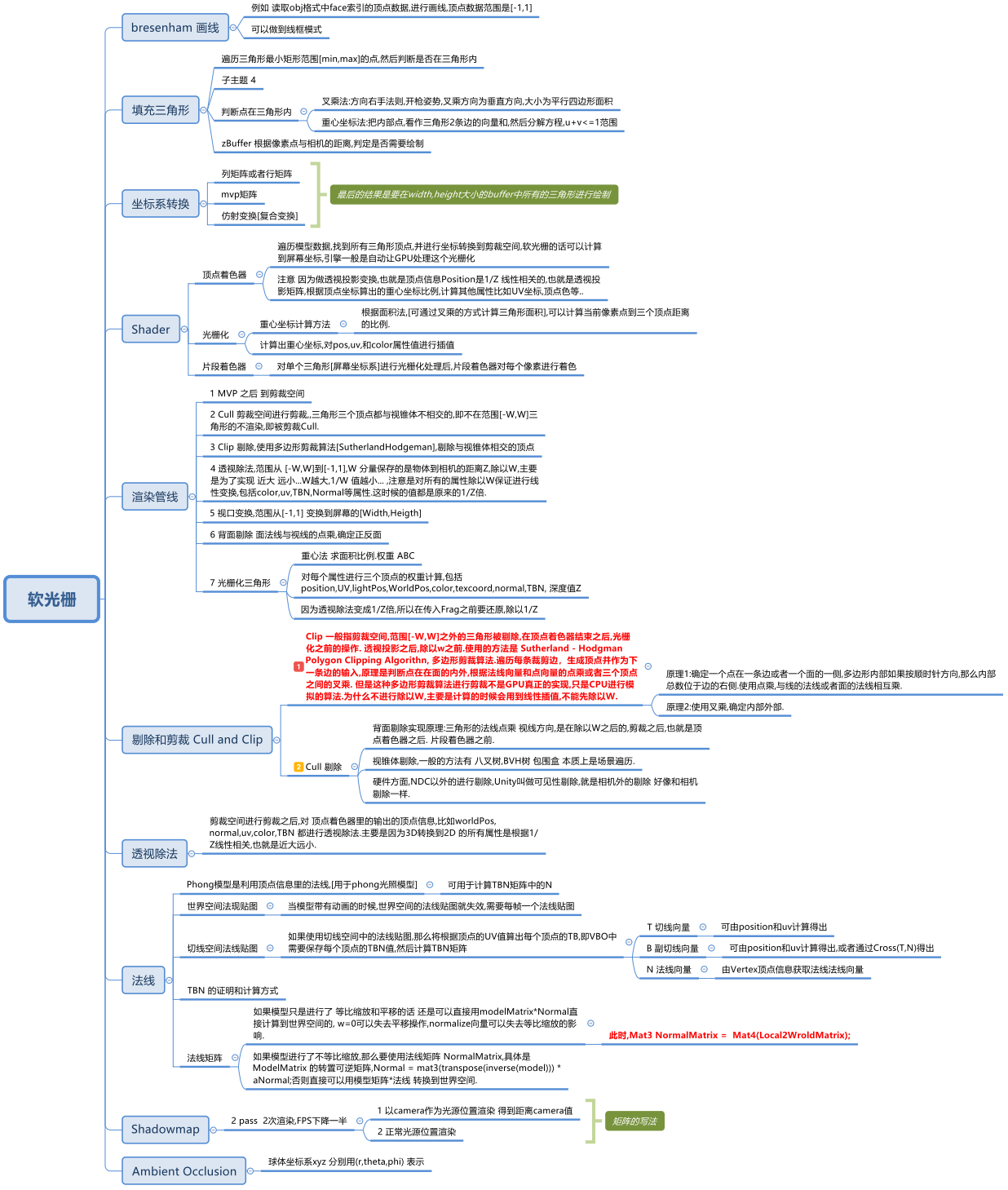
软光栅
bresenham 画线
例如 读取obj格式中face索引的顶点数据,进行画线,顶点数据范围是[-1,1]
可以做到线框模式
填充三角形
遍历三角形最小矩形范围[min,max]的点,然后判断是否在三角形内
子主题 4
判断点在三角形内
叉乘法:方向右手法则,开枪姿势,叉乘方向为垂直方向,大小为平行四边形面积
重心坐标法:把内部点,看作三角形2条边的向量和,然后分解方程,u+v<=1范围
zBuffer 根据像素点与相机的距离,判定是否需要绘制
坐标系转换
列矩阵或者行矩阵
mvp矩阵
仿射变换[复合变换]
最后的结果是要在width,height大小的buffer中所有的三角形进行绘制
Shader
顶点着色器
遍历模型数据,找到所有三角形顶点,并进行坐标转换到剪裁空间,软光栅的话可以计算到屏幕坐标,引擎一般是自动让GPU处理这个光栅化
注意 因为做透视投影变换,也就是顶点信息Position是1/Z 线性相关的,也就是透视投影矩阵,根据顶点坐标算出的重心坐标比例,计算其他属性比如UV坐标,顶点色等..
光栅化
重心坐标计算方法
根据面积法,[可通过叉乘的方式计算三角形面积],可以计算当前像素点到三个顶点距离的比例.
计算出重心坐标,对pos,uv,和color属性值进行插值
片段着色器
对单个三角形[屏幕坐标系]进行光栅化处理后,片段着色器对每个像素进行着色
渲染管线
1 MVP 之后 到剪裁空间
2 Cull 剪裁空间进行剪裁,,三角形三个顶点都与视锥体不相交的,即不在范围[-W,W]三角形的不渲染,即被剪裁Cull.
3 Clip 剔除,使用多边形剪裁算法[SutherlandHodgeman],剔除与视锥体相交的顶点
4 透视除法,范围从 [-W,W]到[-1,1],W 分量保存的是物体到相机的距离Z,除以W,主要是为了实现 近大 远小...W越大,1/W 值越小... ,注意是对所有的属性除以W保证进行线性变换,包括color,uv,TBN,Normal等属性.这时候的值都是原来的1/Z倍.
5 视口变换,范围从[-1,1] 变换到屏幕的[Width,Heigth]
6 背面剔除 面法线与视线的点乘,确定正反面
7 光栅化三角形
重心法 求面积比例.权重 ABC
对每个属性进行三个顶点的权重计算,包括position,UV,lightPos,WorldPos,color,texcoord,normal,TBN, 深度值Z
因为透视除法变成1/Z倍,所以在传入Frag之前要还原,除以1/Z
剔除和剪裁 Cull and Clip
1
Clip 一般指剪裁空间,范围[-W,W]之外的三角形被剔除,在顶点着色器结束之后,光栅化之前的操作. 透视投影之后,除以w之前.使用的方法是 Sutherland - Hodgman Polygon Clipping Algorithm, 多边形剪裁算法.遍历每条裁剪边，生成顶点并作为下一条边的输入,原理是判断点在在面的内外,根据法线向量和点向量的点乘或者三个顶点之间的叉乘. 但是这种多边形剪裁算法进行剪裁不是GPU真正的实现,只是CPU进行模拟的算法.为什么不进行除以W,主要是计算的时候会用到线性插值,不能先除以W.
原理1:确定一个点在一条边或者一个面的一侧,多边形内部如果按顺时针方向,那么内部总数位于边的右侧.使用点乘,与线的法线或者面的法线相互乘.
原理2:使用叉乘,确定内部外部.
2
Cull 剔除
背面剔除实现原理:三角形的法线点乘 视线方向,是在除以W之后的,剪裁之后,也就是顶点着色器之后. 片段着色器之前.
视锥体剔除,一般的方法有 八叉树,BVH树 包围盒 本质上是场景遍历.
硬件方面,NDC以外的进行剔除,Unity叫做可见性剔除,就是相机外的剔除 好像和相机剔除一样.
透视除法
剪裁空间进行剪裁之后,对 顶点着色器里的输出的顶点信息,比如worldPos, normal,uv,color,TBN 都进行透视除法.主要是因为3D转换到2D 的所有属性是根据1/Z线性相关,也就是近大远小.
法线
Phong模型是利用顶点信息里的法线,[用于phong光照模型]
可用于计算TBN矩阵中的N
世界空间法线贴图
当模型带有动画的时候,世界空间的法线贴图就失效,需要每帧一个法线贴图
切线空间法线贴图
如果使用切线空间中的法线贴图,那么将根据顶点的UV值算出每个顶点的TB,即VBO中需要保存每个顶点的TBN值,然后计算TBN矩阵
T 切线向量
可由position和uv计算得出
B 副切线向量
可由position和uv计算得出,或者通过Cross(T,N)得出
N 法线向量
由Vertex顶点信息获取法线法线向量
TBN 的证明和计算方式
法线矩阵
如果模型只是进行了 等比缩放和平移的话 还是可以直接用modelMatrix*Normal直接计算到世界空间的, w=0可以失去平移操作,normalize向量可以失去等比缩放的影响.
此时,Mat3 NormalMatrix = Mat4(Local2WroldMatrix);
如果模型进行了不等比缩放,那么要使用法线矩阵 NormalMatrix,具体是ModelMatrix 的转置可逆矩阵,Normal = mat3(transpose(inverse(model))) * aNormal;否则直接可以用模型矩阵*法线 转换到世界空间.
Shadowmap
2 pass 2次渲染,FPS下降一半
1 以camera作为光源位置渲染 得到距离camera值
2 正常光源位置渲染
矩阵的写法
Ambient Occlusion
球体坐标系xyz 分别用(r,theta,phi) 表示