## LightShaftsCommandBuffer

Middle Quality: 5 blit

前面都是half res，最后upscale

第一步：pass 4 - downsample depth to half

Bilinear Filter了一下depth texture

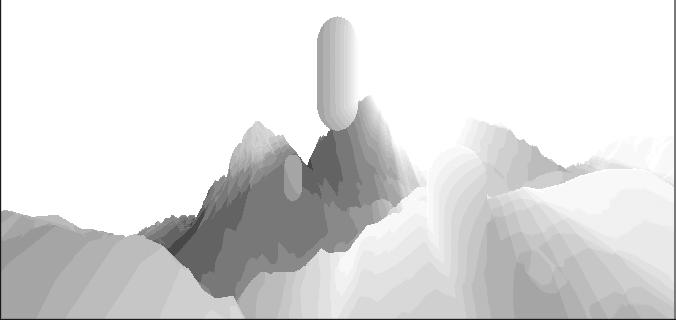
注意到vs里面也做了处理

第二步：pass 10 - light shaft

绘制出inscattering texture

SampleCount设成10次就已经有比较好的效果了

使用了一张\_DitherTexture来扰动raymarch的初始位置，如果去掉这个扰动，则渲染结果会有明显的banding：



当然也会导致最终结果的artifact



第三步：pass 2 - horizontal blur (lores)

第四步：pass 3 - vertical blur (lores)

Blur的对象是inscattering texture( \_MainTex)

与中心点的Depth Diff用来算权重，权重与depth diff成指数级衰减关系

light shaft效果：如果2个点depth接近，则效果会融合到一起（这样就不会有很多很细的丝状了）；如果2个点depth相差较大（边界），则效果不会融合

第五步：pass 5 - bilateral upsample

最终输出的是low res depth里面的值

先求high res depth和周围4个像素的low res depth的差值之和，如果很小则说明not an edge，直接返回当前像素的lodepth；

否则说明这里是个edge，那么则找到前面的4个差值里面最小的那个，返回它的uv对应的lodepth

light shaft效果：非edge的位置充分融合，edge的位置则取最近的blur过的texel

High Quality: 3 blit

自始至终都是full res

第一步：pass 10 - light shaft

第二部：pass 0 - horizontal blur (hires)

第三部：pass 1 - vertical blur (hires)